

sandoba.de medien agentur
contact@sandoba.de / www.sandoba.de

Contentpapst-Bundle 2010.2 - Handbuch für Entwickler

5. Systemkomponenten / Module

6. Erweiterung des Systems per Entwickler-API

1. Installation des Systems

1.1. Übersicht

1.1.0. Installation des Sytems

Zur Installation des Systems beachten Sie bitte die in der Installationsanleitung beschriebenen Schritte. Die Installationsanleitung ist auf der CD-Rom im PDF-Format enthalten und kann alternativ unter http://www.sandoba.de/ heruntergeladen werden.

1.1.1. Systemanforderungen von Contentpapst

Das Content-Management-System **Contentpapst** nutzt ausschließlich im Internet weit verbreitete Technologien und Programmiersprachen. Sie können daher **Contentpapst** auf jedem Webserver benutzen, welcher über eine MySQL-Datenbank und die Unterstützung von PHP 4.3+ verfügt.

PHP ist die Abkürzung für "PHP Hypertext Preprocessor" und eine der wesentlichen Vorraussetzungen für dynamische Inhalte im Internet. Diese Programmiersprache erfreut sich aufgrund einer leicht erlernbaren Struktur hoher Beliebtheit. Mittlerweile ist PHP zusammen mit einer MySQL-Datenbank bereits in vielen preiswerten Webhosting-Paketen enthalten.



Um alle Funktionen von **Contentpapst** nutzen zu können, bietet es sich an, die GD-Library zur dynamischen Generierung von Grafiken zu installieren. Die GD-Library ist eine standardmäßig aktivierte Erweiterung von PHP und kann in **Contentpapst** beispielsweise zur Erstellung von grafischen Navigationsmenüs oder für die dynamische Anpassung der Grafiken verwendet werden.

Bereits im Systemen enthalten sind Funktionen zur Generierung von PDF-Dokumenten. Mit diesen können der Administrator und die Redakteure auf Wunsch einzelne Seitenelemente wie z.B. Artikel oder News als PDF-Datei generieren, auf dem Server speichern lassen und so auch für Print-Medien bereitstellen.

Zusammengefasst sind die Systemanforderungen folgende:

- PHP ab Version 4.3, auch 5.0+ ist möglich
- MySQL-Datenbank mit mindestens 5 MB Speicherplatz (ab Version 3.23)
- optional: die GD-Library für PHP (ab Version 2.01)
- open_basedir = Off, safe_mode = Off (Einstellungen für PHP)
- mind. 10 MB Webspace für die Dateien von **Contentpapst**
- **weiterer Webspace wird je nach Umfang der Nutzung benötigt**

Sie sehen also, **Contentpapst** stellt trotz der umfangreichen Funktionen nur sehr geringe Systemanforderungen und kann daher bei fast jedem Webhoster eingesetzt werden. Eine Liste mit bereits getesteten Webhostern haben wir unter http://www.sandoba.de/support/webhoster-im-test/ zusammengestellt.

1.1.2. Überprüfung der Systemanforderungen

Contentpapst wird direkt auf einem Webserver installiert, d.h. auf dem Server, auf dem Sie auch normale HTML-Seiten lagern. Die Voraussetzung für eine erfolgreiche Installation ist, dass der Webserver über PHP (Programmiersprache, ab Version 4.3) und eine MySQL-Datenbank (ab Version 3.23) verfügen. Dies können Sie überprüfen, indem Sie folgenden Text in eine Datei mit dem Namen "phpinfo.php" kopieren, diese per FTP auf Ihren Webspace laden und über den Browser aufrufen:

```
<?php phpinfo(); ?>
```

Sofern PHP ordnungsgemäß installiert ist, sehen Sie in der Ausgabe des Browsers alle wissenswerten Informationen über Ihren Webserver, Installierte Erweiterungen von PHP uvm. Direkt am Anfang der Ausgabe sehen Sie die Versionsnummer von PHP, für MySQL gibt es einen eigenen Abschnitt. Eine Übersicht zu den benötigten Systemanforderungen der aktuellen Version finden Sie immer unter http://www.sandoba.de/. Wenn Sie eine ältere PHP-Version als 4.3 angezeigt bekommen oder ab die Systemanforderungen anderweitig nicht erfüllt werden, dann kann **Contentpapst** auf Ihrem Webserver nicht installiert werden.

1.1.3. Updates / Upgrades installieren

Um Updates / Upgrades für das System zu installieren, befolgen Sie die in der beiliegenden Anleitung erläuterten Schritte. Diese beinhalten u.a. die Aktualisierung von Dateien per FTP und das Einspielen von SQL-Befehlen über das **Contentpapst**-Modul "Sicherung". Damit lässt sich die Umstellung auf eine neue Version weitgehend automatisieren.

Updates sind u.a. für diese Versionen verfügbar:

- Version 1.31 > Version 1.38
- Version 1.38 > Version 2.00
- Version 2.00 > Version 2.01
- Version 2.00 > Version 2.10
- Version 2.10 > Version 2.50
- Version 2.50 > Version 3.00
- Version 3.00 > Version 3.10
- Version 3.10 > Version 3.50
- Version 3.50 > Version 2007.1
- Version 2007.1 > Version 2008.1
- Version 2008.1 > Version 2008.2
- Version 2008.2 > Version 2008.3
- Version 2008.3 > Version 2009.1
- Version 2009.1 > Version 2009.2
- **ab Version 2009.2 wird ein neues Updatetool genutzt**

Sobald ein neues Update für **Contentpapst** erscheint, erhalten Sie automatisch auf der Startseite des Administrationsbereiches eine entsprechende Meldung eingeblendet. Um das Update zu erhalten, können Sie sich dann an den Support wenden oder Sie laden dieses direkt aus Ihrem Kunden-Konto unter http://www.sandoba.de/shop-login.html herunter.

1.1.4. Vorlage "container/delivery.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/container/" und dient der Anzeige der im Shop verfügbaren Lieferstati.

{cp:tags} für diese Vorlage	
{cp:shop:delivery:}*	Alle Angaben zum Lieferstatus aus der Datenbanktabelle *_shop_delivery.

1.2. Häufige Fragen zur Installation

1.2.1. Wie gehe ich beim Update auf eine neue Contentpapst-Version vor?

Jedem Update von Contentpapst liegt eine ausführliche Anleitung mit Anweisungen bei, die den Installationsprozess genau beschreiben. Abhängig vom Umfang des Updates kann es sich hierbei um Datenbankänderungen, Änderungen an Dateien oder Ordnerstrukturen handeln. Sofern dies möglich ist, wird mit dem Update auch eine Installationsdatei mitgeliefert, welche einen Großteil der Änderungen automatisch durchführt. Benötigen Sie Hilfe bei der Installation eines Updates, setzen Sie sich bitte mit dem Support in Verbindung.

1.2.2. Bei der Installation erscheint "Warning: session_start()...", warum?

Wenn nach dem Hochladen des Systems über den Installer beim Aufruf des Installationsprozesses die folgende Fehlermeldung erscheint, dann müssen Sie vor der Installation noch die Zugriffsrechte (CHMOD) für den Ordner "sessions" manuell über ein beliebiges FTP-Programm auf 755 oder 777 erhöhen:

```
Warning: session_start(): open(./sessions/sess_di6fdsaf, O_RDWR) failed: Permission denied (13) in /is/htdocs/..
Warning: session_start(): Cannot send session cookie - headers already sent out...
```

Dies ist notwendig, damit Contentpapst temporäre Dateien im Session-Ordner anlegen kann. Auch für andere Ordner des Systems kann u.U. eine manuelle Änderung des CHMOD erforderlich sein. Alle Details dazu entnehmen Sie bitte der dem System beiliegenden Installationsanleitung. Ebenfalls wird dort erläutert, wie Sie den CHMOD eines Ordners bzw. einer Datei ändern können.

1.2.3. Bei der Installation funktioniert mein "Hostname" nicht!

Einige Webhoster nutzen für den Zugriff auf den Server per FTP und MySQL verschiedene Hostnamen, z.B. weil die MySQL-Datenbank auf einem anderen Server liegt, als die sonstigen Dateien Ihres Webspaces. Fragen Sie bei Ihrem Webhoster explizit nach dem Hostnamen für die MySQL-Datenbank für Ihre Domain. Da viele Webhoster den Kunden direkt nach der Freischaltung die Zugangsdaten für die MySQL-Datenbank per Email übermitteln bzw. diese in einem Control Panel bereithalten, sollten Sie auch dort die benötigten Angaben finden. Wenn die MySQL-Datenbank dagegen auf dem gleichen Server liegt, dann können Sie in der Regel als Hostnamen "localhost" angeben.

1.2.4. Die Installation wird unterbrochen / ist nicht möglich. Was ist zu tun?

Für jeden Prozess der auf einem Webserver läuft ist allgemein eine maximale Zeit festgelegt, innerhalb der dieser Prozess abgeschlossen werden muss. Bei der Installation ist ein solcher Prozess z.B. das Einrichten der benötigten Datenbankstruktur mit zahlreichen Inhalten wie der Anleitung etc.

Allerdings gibt es bei einigen Webhostern sehr restriktive Zeiteinschränkungen, die es unter Umständen unmöglich machen, die Installation komplett durchzuführen. Dies äussert sich dann in entsprechenden Fehlermeldungen bei der Installation.

Wenn dies bei Ihnen der Fall sein sollte, gehen Sie bitte folgendermaßen vor:

- öffnen Sie die Datei "admin_install.sql" aus dem Hauptverzeichnis in einem beliebigen Text-Editor
- ersetzen Sie dann per "Suchen-und-Ersetzen"-Dialog überall "{CP:SQL}" durch den bei der Installation gewählten Datenbankpräfix sowie "_" (d.h. bei "cp" als Datenbankpräfix wird "cp_" eingesetzt)
- anschließend importieren Sie über ein Tool wie phpMyAdmin jeweils ca. 1000 Zeilen aus dieser Datei in die Datenbank
- zuletzt müssen Sie noch die Datei "admin_install.sql" leeren und leer wieder hochladen

Nachdem nun die Datenbank korrekt mit den Tabellen und Inhalten gefüllt ist, kann die Installation normal fortgesetzt werden, indem Sie einen Administrator für das System einrichten (siehe auch beiliegende Installationsanleitung).

Sollte auch ein Import bspw. per phpMyAdmin aufgrund der Laufzeitbeschränkungen nicht funktionieren, dann ist ein Rückgriff auf ein spezielles Import-Programm wie "BigDump" (http://www.ozerov.de/bigdump.php) möglich. Der Einsatz von BigDump ist aber nur fortgeschrittenen Anwendern mit Datenbankkenntnissen zu empfehlen.

1.2.5. Kann Contentpapst auf Webspace mit aktiviertem "safe_mode" genutzt werden?

Dies ist leider nicht möglich. Die Einstellung "safe_mode" in der PHP-Konfiguration unterbindet den Einsatz verschiedenster wichtiger Funktionen und schränkt vor allem die Möglichkeiten zur Arbeit mit Dateien stark ein (u.a. beim Anlegen, Löschen und Hochladen von Dateien). Somit wäre nicht der volle Funktionsumfang von Contentpapst verfügbar. U.a. sind folgende Funktionen bei einem aktivierten "safe_mode" nicht bzw. nur eingeschränkt nutzbar:

- *copy()* - Kopieren von Dateien
- *mkdir()* - Erstellung eines neuen Ordners
- *rename()* - Umbenennen von Dateien
- *rmdir()* - Vorhandenen Ordner löschen
- *unlink()* - Löschen von Dateien
- ...

"safe_mode" muss also auf Ihrem Webspace deaktiviert sein. Einige Webhoster deaktivieren den "safe_mode" auf Wunsch für den Webspace (da der "safe_mode" jeweils lokal und global eingestellt werden kann). Alternativ halten wir eine Liste mit getesteten Webhostern ("Webhoster im Test") für Sie bereit, so dass Sie sich für einen passenden Webhoster entscheiden können.

1.2.6. Was muss ich beim Umzug auf einen anderen Server beachten?

Wenn Sie auf einen anderen Server umziehen wollen, lesen Sie bitte den Punkt "Wie wird bei einer Deinstallation vorgegangen?" um zu erfahren, wie Sie ein lokales Backup der Datenbank erstellen und anschließend die Datenbank löschen können. Bitte beachten Sie, dass Sie die Dateien, die Sie selbst hinzugefügt haben (wie z.B. Grafiken, PDF-Dateien etc.), vorher per FTP sichern müssen.

Nachdem Contentpapst dann deinstalliert ist, wenden Sie sich per Email an den Support und lassen Sie Ihre Registrierung wieder für die Installation freischalten. Auf dem neuen Server nehmen Sie die Installation erneut vor (mit dem gleichen Datenbankpräfix), die Datei "admin_install.sql" tauschen Sie vorab jedoch mit der Backup-Datei aus. Anschließend laden Sie noch die gesicherten Dateien an die vorherigen Positionen im Dateisystem und schon ist der Umzug abgeschlossen.

1.2.7. Welche Bedeutung hat die PHP-Einstellung "open_basedir"?

Die PHP-Einstellung "open_basedir" beschränkt den Zugriff von PHP-Dateien auf die darunterliegenden Ordnern und Dateien. Bei jedem Zugriff - wie z.B. beim Öffnen einer Datei - wird überprüft, wo diese liegt und ob entsprechende Berechtigungen für das aufrufende Script / den Benutzer vorhanden sind. Ist dies nicht der Fall, dann wird eine Fehlermeldung wie diese zurückgegeben:

```
Warning: open_basedir restriction in effect. File is in wrong directory in /htdocs/files/script.php on line 5
```

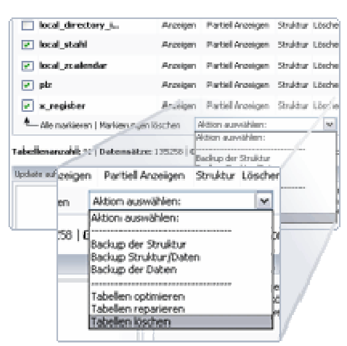
Contentpapst ist so konfiguriert, dass möglichst wenig Operationen durchgeführt werden müssen, die ein deaktiviertes "open_basedir" erfordern. Jedoch lässt sich dies nicht immer vermeiden, z.B. wenn ein dateibasierter Cache oder die Überprüfung externer URLs funktionsfähig sein sollen.

Um "open_basedir" manuell zu deaktivieren, müssen Sie die Datei "php.ini" öffnen (wo diese zu finden ist, ist abhängig vom Betriebssystem des Servers). Dort suchen Sie nach "open_basedir" und setzen dies auf "0" oder "off". Wenn Sie keinen eigenen Server nutzen, so kontaktieren Sie bitte Ihren Webhoster.

1.2.8. Welches Betriebssystem kann von Contentpapst genutzt werden?

Da für Contentpapst keine zusätzliche Software auf dem Server oder einem Client installiert werden muss, ist der Betrieb von Contentpapst prinzipiell auf jedem Betriebssystem möglich, welches als Webserver Apache oder IIS unterstützt. Dazu gehören u.a. Linux (in diversen Distributionen), Unix, Windows 95/98, Windows ME, Windows 2000/2003, Windows XP und Windows Vista.

1.2.9. Wie wird bei einer Deinstallation vorgegangen?



Wenn Sie Contentpapst löschen wollen bzw. zu einem anderen Webhoster umziehen und deshalb die alte Installation löschen müssen (sofern Sie nur eine Lizenz besitzen), müssen Sie zuerst die Datenbanktabellen entfernen. Verwenden Sie hierzu das Modul "Sicherung / Datenbank". Markieren Sie dort alle Datenbanktabellen mit dem bei der Installation gewählten Datenbankpräfix und wählen Sie "Tabellen löschen" aus dem Dropdown-Menü am unteren Ende der Tabelle aus. Die Tabellen werden daraufhin unverzüglich gelöscht.

Achtung! Wenn Sie die Löschung der Datenbanktabellen ohne vorherige Datensicherung veranlassen, sind alle Inhalte unwiderruflich gelöscht!

Um vorab eine Datensicherung durchzuführen, wählen Sie statt "Tabellen löschen" besser "Backup Struktur/Daten" aus dem Dropdown-Menü bzw. "Quick-Backup" aus der oberen Menüleiste aus. Daraufhin startet ein Download mit dem SQL-Backup der Inhalte. Bei einer Deinstallation können Sie nach dem Löschen der Datenbanktabellen die restlichen Dateien einfach per FTP entfernen.

1.2.10. Welche Bedeutung hat die Fehlermeldung "Allowed memory size of ... bytes exhausted"?

Diese Fehlermeldung kann u.U. bei der Installation oder anderen besonders speicherintensiven Prozessen auftreten, bei denen größere Dateien geladen und verarbeitet werden müssen. Eine Fehlermeldung im unten sichtbaren Format weist darauf hin, dass für diesen Webspace ein zu geringer Speicher (RAM) zugewiesen ist. In diesem Beispiel stehen für PHP nur maximal 8MB zur Verfügung:

```
"Fatal error: Allowed memory size of 8388608 bytes exhausted (tried to allocate 4279205 bytes)
in /homepages/files/install.php on line 1205"
```

Hier gibt es zwei mögliche Vorgehensweisen: Wenn es sich um einen eigenen Server handelt, dann kann in der Konfigurations-Datei "php.ini" der Wert "memory_limit" erhöht werden (z.B. auf "32M" für 32 MB RAM). Ohne Zugriff auf diese Datei kann - wenn das Problem direkt bei der Installation auftritt - die Installation nach den in "Die Installation wird unterbrochen / ist nicht möglich. Was ist zu tun?" beschriebenen Schritten fortgesetzt werden.

1.2.11. Welche Bedeutung hat die Fehlermeldung "Maximum execution time ... exceeded"?

Diese Fehlermeldung bedeutet, dass der Server nach z.B. 30 Sekunden die Ausführung eines Prozesses abgebrochen hat, da auf diesem Server dann 30 Sekunden als maximale Laufzeit für PHP-Scripte festgelegt wurden. Das System versucht dies automatisch zu erhöhen. Da auf manchen Servern eine Erhöhung durch ein PHP-Script jedoch ebenfalls nicht erlaubt wird, kann dann eine Installation nach den in der Anleitung unter "Die Installation wird unterbrochen / ist nicht möglich. Was ist zu tun?" beschriebenen Schritten durchgeführt werden.

1.2.12. Wo erhalte ich meinen Benutzernamen und mein Passwort?

Scheinbar wurde die Installation nicht komplett zuende geführt, denn im letzten Schritt der Installation werden Sie zur Eingabe von wichtigen Daten für den Administrator aufgefordert. Dies sind u.a. Email-Adresse, Benutzername und Passwort.

Wenn nun keine Möglichkeit mehr besteht, zur Installation zurückzukehren, dann müssen Sie über ein Datenbanktool wie z.B. phpMyAdmin (http://www.phpmyadmin.net/) in die Datenbanktabelle *_user wechseln ("*" ist durch den gewählten Datenbankpräfix zu ersetzen). Dort ist folgender SQL-Befehl zu importieren (auch hier wieder "*" ersetzen):

```
INSERT INTO *_user (`forename`, `nickname`, `country`, `email`, `username`, `password`, `gender`, `level`,
`clanguage`, `showemail`, `sendmail`) VALUES ('Max', 'Mustermann', '54', 'ihre@email.de', 'admin',
'd56b699830e77ba53855679cb1d252da', 'm', 999, 'de', '1', '1');
```

Sie können sich dann unter "index.php?file=user" mit dem Benutzernamen "admin" und dem Passwort "login" einloggen und anschließend im Administrationsbereich Ihre Daten anpassen.



ACHTUNG: Sie sollten die Daten des neu angelegten Administrators möglichst sofort bearbeiten, um Sicherheitslücken zu vermeiden. Insbesondere sind der Benutzername und das Passwort zu ändern.

2. Erste Schritte mit dem System

In Zeiten in denen der digitale Geschäftsverkehr immer präsenter wird und überall von Webservices der Zukunft die Rede ist, benötigen speziell kleine und mittelständische Unternehmen leistungsstarke Programme, die ihre tägliche Arbeit erleichtern. ***Contentpapst*** *ist ein solches Programm für den Online-Bereich. Entdecken Sie die Möglichkeiten, welche sich durch* ***Contentpapst*** *eröffnen und Ihnen Zeit und damit Geld sparen helfen. Den Content-Management-Systemen gehört die Zukunft!*

2.1. Übersicht

2.1.0. Was ist ein Content-Management-System (CMS)?

CMS ist die Abkürzung für "Content-Management-System" und bedeutet übersetzt in etwa die Verwaltung und Organisation von Inhalten innerhalb eines Systemes. Ein CMS erlaubt genau dies im Internet bzw, in externen Netzen (Extranet) oder geschlossenen Netzen (Intranets).

Bisher wurden Webseiten meist in einem Editor erstellt. Anschließend musste man die generierten Dateien aufwändig auf einen Server laden, bei kleinen Änderungen viele Dateien anpassen und nicht zuletzt einiges über Technologien im Internet usw. wissen.

Dies ändert sich mit der Hilfe von Content-Management-Systemen gewaltig. Die sandoba.de medien agentur ist einer der ersten Anbieter, welcher mit **Contentpapst** eine Komplettlösung auf dem Markt der CMS vorstellt. **Contentpapst** wurde speziell für die Ansprüche von klein- und mittelständischen Unternehmen, sowie privater Webseiten entwickelt. Das System vereint die Vorteile neuester Technologien mit einem flexiblen Back- und Frontend und ermöglicht es, Webseiten ohne Programmierkenntnisse und mit geringen Kosten zu bearbeiten.

2.1.1. Warum Contentpapst als CMS?

Im Gegensatz zu vielen anderen Content-Management-Systemen erhalten Sie mit **Contentpapst** eine "All-Inclusive"-Lösung auf einer soliden technischen Basis und mit glänzenden Zukunftsaussichten. Funktionen, welche bislang nur bei Systemen für vier- bis fünfstellige EUR-Beträge zu finden waren, gehören bei **Contentpapst** bereits zur Grundausstattung.

Das System kann so z.B. Inhalte der Webseite in verschiedenen Sprachen verarbeiten. Jeder Benutzer kann zwischen den angebotenen Sprachen frei wechseln und auch der Administrationsbereich passt sich der Sprache des Benutzers an. So können Sie **mehrsprachige Versionen Ihrer Webseite** in einem Layout erstellen und unterschiedliche Zielgruppen schnell und günstig informieren. Die Module zur Integration der Sprachen können auf http://www.sandoba.de/ zusätzlich erworben werden und lassen sich einfach installieren.

Der **WYSIWYG-Editor** (What You Se Is What You Get) erlaubt die intuitive Änderung von Datei-Inhalten, Artikeln und Vorlagen anhand von aus Textbearbeitungsprogrammen bekannten Funktionen. Sehen Sie alle vorgenommenen Änderungen in Echtzeit in einem Editor. Profis können weiterhin HTML-Code und Scripte in getrennten Editoren einfügen.

Ein **Gruppen-Kalender** mit einer Aufgabenübersicht zeigt jederzeit den Redakteuren genau an, welche Aufgaben umgesetzt werden müssen. Zusammen mit der **zeitlich gesteuerten Veröffentlichung von Inhalten** können Sie somit genau bestimmen, wie sich Ihre Webseite präsentiert.

Zur Administration Ihrer Webseite sind **keine PlugIns** notwendig. Das System kann kann komplett über den Browser bedient werden. Weiterhin fallen **keine Kosten durch zusätzliche Arbeitsplatz-Lizenzen** an. **Contentpapst** kann von beliebig vielen Redakteuren genutzt werden. Im Gegensatz zu anderen Systemen müssen keine teuren Zusatz-Lizenzen erworben werden. Sie statten die Nutzer des Systemes mit spezifischen Rechten aus, welche einem Bildredakteur andere Möglichkeiten einräumen, als z.B. dem Chefredakteur.

Funktionen zur Automatisierung von Prozessen und die Möglichkeit Inhalte der Webseite jederzeit von jedem Ort zu ändern steigern die Produktivität und Rentabilität Ihrer Webseite enorm. Zudem erlaubt die Automatisierung eine Dynamisierung in der Darstellung der Webseite. Sie sind flexibel und an keine festgefahrenen Strukturen gebunden.

2.1.2. Copy & Paste

Das System von **Contentpapst** ist offen für textbasierte Inhalte aus anderen Anwendungen. Sie können diese Inhalte aus Textbearbeitungsprogrammen, WYSIWYG-Editoren usw. einfach herauskopieren und in **Contentpapst** direkt einfügen, ohne erst Dateien hochladen zu müssen. Für die WYSIWYG-Editoren gibt es (in der erweiterten Ansicht) zusätzlich einige Funktionen um per Copy & Paste eingefügte Inhalte zu säubern (damit **Contentpapst** eine valide Ausgabe generieren kann).

2.1.3. Passwortverschlüsselung per MD5/SHA1-Algorithmus

Alle sensitiven Daten der Mitglieder, Redakteure und Administratoren sind durch Benutzername und Passwort gesichert. Das Passwort wird zusätzlich mittels des MD5-Algorithmus verschlüsselt und liegt auch in der Datenbank nicht in entschlüsselter Form vor. Alternativ zum MD5-Algorithmus kann der SHA1-Algorithmus verwendet werden. Dies lässt sich über die Einstellungen unter "Allgemeine Einstellungen" aktivieren.

Beide Algorithmen bieten höchste Sicherheit, bisher ist eine Entschlüsselung nicht möglich. Für vergessene Passwörter bietet **Contentpapst** eine Passwort-Recovery-Funktion an. Dem Benutzer wird dazu ein neu generiertes Passwort per Email zugesandt.

2.1.4. Newsfeed im RSS/RDF-Format

Auch im RSS/RDF-Format lassen sich einzelne Inhalte Ihrer Webseite automatisch generieren. Dieses Format dient u.a. zum Austauschen von Inhalten und zum Einsatz auf weiteren Webseiten. Im Gegensatz zum Newsfeed für KlipFolio sind die Möglichkeiten hier vielfältiger.

Die Daten werden in der RSS/RDF-Formatierung als XML-Datei erstellt und beschreiben den Inhalt der jeweiligen Newsbeiträge. Alle Daten können beim Einbinden auf anderen Webseiten frei formatiert werden, da sie unabhängig von jeglicher Formatierung gespeichert werden. Die XML-Datei für die News-Einträge finden Sie im Hauptordner als "newsfeed.xml".

2.1.5. Übernahme bestehender Webseiten

Die Übernahme bestehender Webseiten ist mit **Contentpapst** keineswegs mit viel Arbeit verbunden. Richten Sie hierfür zuerst ein neues Vorlagen-Set ein, in welches Sie den vorhandenen HTML-Code integrieren. Wichtig sind hierbei Vorlagen für die gesamte Seite sowie die Aufteilung der Inhalte in Tabellen und einzelnen Unterseiten.

Laden Sie dann Bilder und andere benötigte Dateien in Unterverzeichnisse Ihres Vorlagen-Sets mittels der Dateiverwaltung hoch und treffen Sie die gewünschten Einstellungen am System. Nun können Sie mit dem Einfügen der Inhalte als News, Artikel oder einzelne Dateien beginnen. Speziell Webseiten mit einer klaren Gliederung können schnell in **Contentpapst** übernommen werden.

3. Gestaltung von Vorlagen

Die komplette Gestaltung der Webseite kann mittels der Vorlagen festgelegt werden. Stellen Sie hier die Farben und sonstige Eigenschaften der einzelnen Seitenelemente zusammen. Vorlagen trennen die technischen Grundlagen des Systems (die Logik) von grafischen und sonstigen Inhalten (der Gestaltung).

3.1. Übersicht

3.1.0. Alternative Seitenlayouts für einzelne Bereiche

Einzelne Bereiche können in **Contentpapst** mit einem komplett anderen Seitenlayout dargestellt werden. Die gewünschte Seite muss dazu mit dem Parameter "set_template" aufgerufen werden. D.h. wenn der Artikel mit der ID "84" im Seitenlayout "alternative.htm" aufgerufen werden soll (diese Datei liegt im Ordner des verwendeten Vorlagen-Sets und ist mit der "main.htm" Vorlage vergleichbar) muss dieser per Link wie folgt aufgerufen werden:

```
*/index.php?id=84&set_template=alternative
```

Aus dem Parameter "set_template" wird der Dateiname zusammengesetzt und um die Dateiendung ".htm" erweitert. Ist diese Datei vorhanden, wird sie zur Darstellung des Layouts genutzt, andernfalls kommt das Standardlayout zum Einsatz.

3.1.1. Statische Seiten in Contentpapst einbinden

Speziell bei älteren Projekten existieren oftmals Daten in unterschiedlichen Formaten, welche z.B. in Form von statischen HTML-Dateien vorliegen. Generell empfiehlt es sich, diese Inhalte in **Contentpapst** zu importieren, um auf erweiterte Funktionen wie die Rechteverwaltung, die Volltextsuche oder Caching-Verfahren zurückgreifen zu können.

Sollte dies jedoch nicht gewünscht werden, können die Vorteile des Vorlagen-Systems von **Contentpapst** dennoch genutzt werden. Hierzu wird die statische Seite über die Datei "index.php" mittels des Parameters "static" aufgerufen und durch **Contentpapst** verarbeitet.

In der HTML-Datei können alle in **Contentpapst** vorhandenen {cp:tags} genutzt werden. Dies erlaubt u.a. die Einbettung von Artikeln, die Anzeige von Forenbeiträgen, eines Datums oder die Anzeige bestimmter Bereiche nur für Mitglieder. Über den optionalen Parameter "clean" lässt sich die Anzeige des allgemeinen Seitenlayouts ein- und ausschalten.

Alle HTML-Dateien müssen aus Sicherheitsgründen innerhalb des **Contentpapst**-Verzeichnisses (bzw. in einem der Unterverzeichnisse) abgelegt werden.

```
*/index.php?static=faq.htm
```

Ruft die Datei "faq.htm" aus dem Hauptverzeichnis auf und verarbeitet alle in dieser Datei vorhandenen {cp:tags}. Die gesamte Datei wird in das Layout der Webseite integriert.

```
*/index.php?static=lokalinformation/steuerabgaben.htm&clean=yes
```

Über diese URL wird die Datei "steuerabgaben.htm" aus dem Verzeichnis "lokalinformation" aufgerufen. Auch hier werden die enthaltenen {cp:tags} verarbeitet, eine Integration in das Layout der Webseite unterbleibt jedoch. Dies erlaubt es, ein komplett eigenständiges Layout aufzubauen.

3.1.2. Stylesheets anpassen

Wesentliche Teile der grafischen Gestaltung werden über die CSS-Stylesheets gesteuert. Diese sind in den Dateien "standard.css" (allgemeine Angaben), "style.css" (CSS für Anzeige per Browser) und "print.css" (CSS für Druckversion) des jeweils genutzten Vorlagen-Sets integriert. Sie legen den genutzten Schrifttyp mit einer Standardgröße und -farbe, sowie die

Gestaltung der Scrollleisten fest. Darüberhinaus werden alle Formular-Elemente wie Eingabefelder, Checkboxen etc. durch die Stylesheets formatiert, Grafiken eingebunden etc.

Eine Umgestaltung aller Textboxen beispielsweise kann dann zentral in dieser Datei vorgenommen werden und gilt für alle Bereiche, welche dieses Vorlagen-Set einsetzen. Auch Navigationselemente werden in der Datei "standard.css" gestaltet. CSS-Angaben, welche im Vorlagen-Set bereits genutzt werden, sollten nicht entfernt werden. Hier ist nur eine Anpassung an das Layout der Webseite ratsam.

3.1.3. Vorlagen "article_*.htm"

"article_*.htm" sind Vorlagen für den grundsätzlichen Aufbau von Artikeln. Die wichtigsten Tags sind hierbei {cp:header} (Titel) und {cp:content} (Inhalt). Ergänzt werden können diese durch die Druckfunktion ({cp:print}), Datumsangaben sowie Bewertungen. Es bietet sich bei den Artikeln an, zwischen Besucher und Mitglied zu unterschieden (mittels {cp:ismember}/{cp:isvisitor}). So können Sie Ihren Mitgliedern zusätzliche Inhalte anbieten. Unterschiedliche Gestaltungsarten lassen sich durch die Anlage verschiedener Dateien nach dem Muster "article_*.htm" realisieren. Bei der Erstellung eines Artikels wird einfach die gewünschte Vorlage ausgewählt.

{cp:tags} für diese Vorlage	
{cp:article:convert}	Suchmaschinenfreundliche URL zum Artikel
{cp:article:date}	Datum der Erstellung
{cp:article:number}	Nummer des Artikels
{cp:break}	Seitenumbruch im Artikel (in Kombination mit {cp:pages} zu nutzen)
{cp:content}	Inhalt des Artikels
{cp:header}	Titel des Artikels
{cp:level}	Navigation für die Anzeige der Position des Artikels innerhalb der Kategorienstruktur
{cp:pages}	Navigation zum Wechsel auf die weiteren Seiten (sofern vorhanden)
{cp:pdf}	Link zur PDF-Version (per Button, PDF-Version wird als PopUp geöffnet)
{cp:print}	Link zur Druckversion (per Button, Druckversion wird als PopUp geöffnet)
{cp:rating}	Grafische Anzeige der Artikelbewertung
{cp:vote}	Formular zur Bewertung des Artikels (muss durch {cp:voteset} komplettiert werden)
{cp:voteset}	Abschluss des Formulares zur Bewertung des Artikels (Button zum Abschicken)

3.1.4. Vorlage "articles_short.htm"

Die Vorlage "articles_short.htm" kommt als Standard-Vorlage für den Container {cp:container module="article"} zum Einsatz, sofern über den Parameter "template" keine andere Vorlage angegeben wurde. Über diesen Container lassen sich z.B. Übersichtsseiten über neu eingetroffene Artikel generieren.

{cp:tags} für diese Vorlage	
{cp:article:content}	Inhalt des Artikels
{cp:article:convert}	Suchmaschinenfreundliche URL zum Artikel
{cp:article:date}	Datum der Erstellung
{cp:article:header}	Titel des Artikels
{cp:article:hits}	Bisherige Aufrufe
{cp:article:link}	Link zur Detailansicht des Artikels
{cp:article:number}	Nummer des Artikels

3.1.5. Vorlage "gallery_list.htm"

In der Detailansicht ("gallery_list.htm") eines Bildes in der Galerie kann die Anzeige durch zahlreiche Tags beeinflusst werden. Alle zu einem Bild gespeicherten Angaben lassen sich hier einzeln mittels der Tags ausgeben und somit auch in komplexe Layouts einbetten. Mittels {cp:print} bzw. {cp:pdf} bietet man den Besuchern die Ausgabe des Bildes in einer Druck- bzw. PDF-Version an.

Weitere Verfeinerungen der Darstellung sind mit {cp:ismember}/{cp:isvisitor} möglich. In Kombination mit {cp:vote}/{cp:voteset} lässt sich so beispielsweise die Bewertung der Bilder auf eingeloggte Mitglieder einschränken.

{cp:tags} für diese Vorlage	
{cp:gallery:date}	Datum der Eintragung
{cp:gallery:image}	Ausgabe des Bildes (komplett als HTML-Tag)
{cp:gallery:level}	Navigation für die Anzeige der Position des Bildes innerhalb der Kategorienstruktur
{cp:gallery:navigation}	Auflistung der verfügbaren Kategorien
{cp:gallery:poster}	Name des Autors (samt Link zum Profil)
{cp:gallery:rating}	Bewertung (als Grafik)
{cp:gallery:text}	Kommentar zum Bild
{cp:gallery:title}	Titel des Bildes
{cp:gallery:views}	Anzahl der bisherigen Aufrufe
{cp:gallery:vote}	Formular zur Bewertung des Bildes (muss durch {cp:voteset} komplettiert werden)
{cp:gallery:voteset}	Abschluss des Formulares zur Bewertung des Bildes (Button zum Abschicken)
{cp:pdf}	Link zur PDF-Version (per Button, PDF-Version wird als PopUp geöffnet)

{cp:print}	Link zur Druckversion (per Button, Druckversion wird als PopUp geöffnet)

3.1.6. Vorlage "gallery_misc.htm"

Die Vorlage "gallery_misc.htm" wird nur für die Galerie benötigt und wird dort für die Gestaltung der verschiedenen Bereiche verwendet, welche durch die Vorlage "gallery_site.htm" nicht erfasst werden. U.a. ist dies der Bereich zur Verwaltung der Bilder durch die Nutzer.

{cp:tags} für diese Vorlage	
{cp:content}	Inhalt des Galerie-Modules (von der URL abhängig)
{cp:gallery:navigation}	Auflistung der verfügbaren Kategorien

3.1.7. Vorlage "gallery_short.htm"

Um in der Galerie eine schnellere Übersicht zu gewährleisten, gibt es die Vorlage "gallery_short.htm". In dieser wird das Thumbnail des Bildes (eine verkleinerte Version) eingebunden samt einiger Zusatzinformationen. Dies können z.B. der Titel des Bildes, sowie die aktuelle Bewertung, das Datum oder der Name der zugeordneten Kategorie sein.

{cp:tags} für diese Vorlage	
{cp:gallery:category}	Name der zugeordneten Kategorie
{cp:gallery:date}	Datum der Eintragung
{cp:gallery:image}	Thumbnail des Bildes (komplett als HTML-Tag)
{cp:gallery:poster}	Name des Autors (samt Link zum Profil)
{cp:gallery:rating}	Bewertung (als Grafik)
{cp:gallery:title}	Titel des Bildes
{cp:gallery:views}	Anzahl der bisherigen Aufrufe

3.1.8. Vorlage "gallery_site.htm"

"gallery_site.htm" stellt den Grundaufbau der Galerie dar. Eingebettet in diese Vorlage sind die Navigation sowie das Menü zur Sortierung und für die Verwaltung von eigenen Bildern durch die Benutzer. Mit {cp:pages} kann zwischen den einzelnen Seiten navigiert werden. Für {cp:content} werden beim Aufruf der Seite die Inhalte eingesetzt.

{cp:tags} für diese Vorlage	
{cp:content}	Inhalt des Galerie-Modules (von der URL abhängig)
{cp:gallery:level}	Navigation für die Anzeige der Position des Bildes innerhalb der Kategorienstruktur
{cp:gallery:menu}	Auflistung der verfügbaren Kategorien
{cp:gallery:navigation}	Anzeige der Hauptmenüpunkte (für die Sortierung der Bilder usw.)
{cp:gallery:sum}	Bilderanzahl in der Kategorie / der gesamten Galerie
{cp:gallery:title}	Titel der jeweiligen Seite (von der URL abhängig)
{cp:pages}	Navigation zum Seitenwechsel (z.B. "1 2 3 weiter")

3.1.9. Vorlage "guestbook_list.htm"

Die Vorlage "guestbook_list.htm" beschreibt den Aufbau eines Gästebucheintrages und gliedert sich im Wesentlichen in die {cp:tags} für Kopf und Inhalt eines Eintrages. Ergänzt werden diese durch Tags für Zeitangaben, die Email-Adresse, die IP-Nummer, den Link zur Webseite usw. Sollten Sie eine Angabe nicht verwenden wollen, so entfernen Sie diese aus der Vorlage oder ändern Sie die Einstellungen des Gästebuch-Modules.

{cp:tags} für diese Vorlage	
{cp:guest:content}	Inhalt des Gästebucheintrages
{cp:guest:date}	Datum der Eintragung
{cp:guest:email}	Emailadresse des Verfassers
{cp:guest:header}	Titel des Gästebucheintrages
{cp:guest:homepage}	Homepage des Verfassers (sofern vorhanden)
{cp:guest:ip}	IP-Adresse des Verfassers
{cp:guest:location}	Herkunftsland des Verfassers

3.1.10. Vorlage "guestbook_site.htm"

Mit der Gästebuchübersicht lässt sich der grundlegende Aufbau des Gästebuches verändern. Fügen Sie zusätzliche Grafiken oder Texte hinzu, formatieren Sie die Darstellung der Einträge oder positionieren Sie Angaben wie den Counter neu.

{cp:tags} für diese Vorlage	
{cp:guest:addentry}	Formular zum Hinzufügen eines neuen Eintrages

{cp:guest:content}	Inhalt des Gästebuches mit den einzelnen Beiträgen
{cp:guest:counter}	Anzahl der bisherigen Aufrufe des Gästebuches

3.1.11. Vorlage "links_list.htm"

Die Vorlage "links_list.htm" beschreibt den Aufbau eines Linkeintrages und gliedert sich in die {cp:tags} für Kopf, Inhalt und Fuß eines Eintrages.

{cp:tags} für diese Vorlage	
{cp:links:content}	Kurzfassung der Beschreibung des Links
{cp:links:footer}	Fussleiste mit Angaben zum Erstellungsdatum, Bewertung usw.
{cp:links:header}	Titel des Links

3.1.12. Vorlagen "include_*.htm"

Wenn Sie innerhalb eines Modules, z.B. bei den Artikeln, den WYSIWYG-Editor verwenden, haben Sie zusätzlich ein Auswahlfeld "Text-Baustein". Über dieses Feld können Sie häufig benötigte Textelemente einfach per Klick einfügen. Damit dies funktioniert, müssen diese Text-Bausteine jedoch erst einmal angelegt werden.

Hierzu legen Sie im verwendeten Vorlagen-Set eine Datei mit einem Dateinamen im Format "include_*.htm" an. Eine Datei "include_copyright.htm" wird dann als "Copyright (include_copyright.htm)" zur Auswahl bereitgestellt. Dort können dann die gewünschten Texte platziert werden (bspw. Copyright-Informationen, Autoren...).

{cp:tags} für diese Vorlage
Keine {cp:tags} in diesen Vorlagen möglich.

3.1.13. Vorlage "links_section.htm"

Die Vorlage "links_section.htm" wird nur für das Link-Modul benötigt und beinhaltet die grafische Darstellung der einzelnen Kategorien. Damit werden sowohl Haupt- als auch jegliche Unterkategorien in der Gestaltung bestimmt.

{cp:tags} für diese Vorlage	
{cp:links:content}	Titel der Kategorie samt Anzahl der enthaltenen Links
{cp:links:header}	Auflistung der Unterkategorien samt Links

3.1.14. Vorlage "links_site.htm"

"links_site.htm" stellt den Grundaufbau des Link-Modules dar. Eingebettet in diese Vorlage sind die Navigation sowie der eigentliche Inhalt. Für {cp:content} werden beim Aufruf der Seite die Inhalte eingesetzt. Neben diesem Grundaufbau kann jede Kategorie eine eigene Vorlage nutzen. Hierzu legt man neue Vorlagen mit dem Namen "links_cat_*.htm" in dem jeweiligen Vorlagen-Ordner an. "*" muss durch den Namen der Vorlage ersetzt werden.

{cp:tags} für diese Vorlage	
{cp:links:content}	Inhalt des Link-Modules (von der URL abhängig)
{cp:links:level}	Navigation für die Anzeige der Position der Links innerhalb der Kategorienstruktur
{cp:links:navigation}	Anzeige der Hauptmenüpunkte (Neueste Links, Favoriten usw.)

3.1.15. Vorlage "main.htm"

Die Hauptseite eines Vorlagen-Sets steuert die grundlegende Darstellung der Webseite. In diese werden mittels der {cp:tags} im Nachhinein alle Inhalte eingebettet. Die Gestaltung ist dementspechend frei und unterliegt keiner Eingrenzung durch interne oder externe Scripte. Durch die Tags legen Sie hier sogenannte "Container" für Navigationsmenüs, Werbeplätze, Umfragen, Artikel usw. an, welche später dynamisch mit Inhalten gefüllt werden. Ausführliche Informationen über die möglichen Tags finden sich in der Anleitung unter "{cp:tags}".

Für den Inhaltsbereich müssen Sie in dieser Vorlage den Tag "{cp:content}" einmalig einfügen. Zusätzlich sollten Sie die CSS-Datei "standard.css" innerhalb des Ordners des Vorlagen-Sets einbinden, damit Formulare etc. auf der Webseite korrekt dargestellt werden können.

{cp:tags} für diese Vorlage	
{cp:content}	Lädt den über die URL bestimmten Inhalt in die Vorlage.

3.1.16. Vorlage "news_text.htm"

In der Vorlage "news_text.htm" können Anpassungen für die standardmäßige Darstellung von News ohne Bild vorgenommen werden. Aktuell werden Tags für Titel, Text, den ausführlichen Text, den Namen des Verfassers und das

Datum unterstützt.

{cp:tags} für diese Vorlage	
{cp:news:date}	Datum der Erstellung
{cp:news:footer}	Fussleiste mit Angaben zum Autor, der Anzahl der Kommentare usw.
{cp:news:friend}	Link zum Formular für die Weiterempfehlung der News
{cp:news:full}	Inhalt der News (Ausführlicher Text)
{cp:news:link}	Link zur Detailansicht der News
{cp:news:number}	Nummer (ID) der News
{cp:news:poster}	Name des Autors (samt Link zum Profil)
{cp:news:text}	Inhalt der News (Text)
{cp:news:title}	Titel der News
{cp:pdf}	Link zur PDF-Version (per Button, PDF-Version wird als PopUp geöffnet)
{cp:print}	Link zur Druckversion (per Button, Druckversion wird als PopUp geöffnet)

3.1.17. Vorlage "news_image.htm"

Die Vorlage für News mit Bild nutzt die gleichen Tags wie die News ohne Bild, jedoch kann hier mit {cp:news:image} und {cp:news:imagefile} zusätzlich das zu nutzende Bild eingebaut werden.

{cp:tags} für diese Vorlage (zusätzlich zur Vorlage "news_text.htm")	
{cp:news:image}	Ausgabe der Grafik (samt <img>-Tag)
{cp:news:imagefile}	Ausgabe des Dateinamens der Grafik

3.1.18. Vorlage "news_detail.htm"

In der Detailansicht wird die komplette News samt aller Links und Kommentar-Möglichkeiten angezeigt. Zusätzlich lassen sich durch die entsprechenden Tags einzelne Abschnitte und Funktionen nur für Mitglieder oder Besucher freischalten (per {cp:ismember} / {cp:isvisitor}). Über {cp:friend} kann die News weiterempfohlen und über {cp:print} ausgedruckt werden.

{cp:tags} für diese Vorlage (zusätzlich zur Vorlage "news_text.htm")	
{cp:news:image}	Ausgabe der Grafik (samt <img>-Tag)
{cp:news:imagefile}	Ausgabe des Dateinamens der Grafik
{cp:news:link1}… {/cp:news:link1}	Anzeige des Link 1
{cp:news:link2}… {/cp:news:link2}	Anzeige des Link 2
{cp:news:link3}… {/cp:news:link3}	Anzeige des Link 3

3.1.19. Vorlage "print.htm"

Viele Inhalte können direkt in einem druckfreundlichen Format ausgegeben werden. Die Vorlage "print.htm" wird hierfür mit Daten gefüllt, welche vorher über den Tag {cp:print} übermittelt wurden. Es sollte eine Breite von 600 Pixeln bei der Gestaltung der Vorlage nicht überschritten werden, damit beim Druck alle Inhalte korrekt ausgegeben werden können.

{cp:tags} für diese Vorlage	
{cp:content}	Inhalt der Druckversion (ein Artikel, eine News etc.).
{cp:header}	Titel der Druckversion (z.B. Titel des Artikels).
	URL der Seite, die auf diese Druckversion geleitet hat (Referer).

3.1.20. Vorlage "start.htm"

Über die Vorlage "start.htm" eines Vorlagen-Sets wird die Startseite gestaltet. Eine derartige Startseite sollte für die Besucher möglichst einen schnellen Überblick über die angebotenen Inhalte sowie u.a. auch eine Auflistung der neuesten Inhalten / News bieten. Dies kann durch den Einsatz der {cp:tags} realisiert werden.

3.1.21. Vorlage "table.htm"

Die Vorlage "table.htm" dient zur Anpassung der allgemein auf der Webseite genutzten Tabellen. Alle Tabellen welche nicht über eine externe Vorlage angepasst werden können, werden auf der Basis dieser Datei aufgebaut. U.a. sind dies Teile des Mitgliedersystems. Innerhalb der Vorlage sind die Tags "{cp:header}" (für den Titel der Tabelle) und "{cp:content}" (für den Inhalt) zu verwenden.

{cp:tags} für diese Vorlage	
{cp:content}	Inhalt der Tabelle

{cp:header}	Titel der Tabelle

3.1.22. Vorlagen für die Vollextsuche

Für die Volltextsuche werden die Vorlagen unter "/modules/search/templates" genutzt. Dies sind standardmäßig "search_article.html" (Artikel), "search_news.html" (News) und "search_shop.html" (Shop-Artikel). Die Links zu den jeweiligen Detailseiten sind dort folgendermaßen zu setzen:

Vorlage "search_article.html":
Mit "mod_rewrite": "article-{cp:search:number}-{cp:rewrite text="{cp:search:title}"}.html"
Ohne "mod_rewrite": "index.php?id={cp:search:number}"

Vorlage "search_news.html":
Mit "mod_rewrite": "news-detail-{cp:search:number}.html"
Ohne "mod_rewrite": "index.php?file=news&mode=detail&number={cp:search:number}"

Vorlage "search_shop.html" (nur mit dem Modul CP::Shop):
Mit "mod_rewrite": "shop-{cp:search:number}-{cp:rewrite text="{cp:search:title}"}.html"
Ohne "mod_rewrite": "index.php?file=shop&mode=details&number={cp:search:number}"

3.1.23. Vorlagen in HTML-Editoren erstellen

Durch das System der {cp:tags} und die konsequente Trennung von Inhalt und Layout können die Vorlagen, welche in **Contentpapst** eingesetzt werden, mit jedem beliebigen HTML-Editor bearbeitet werden. Somit können Sie auf Wunsch auch im gewohnten Umfeld das Layout Ihrer Webseite definieren und dann in **Contentpapst** einbinden.

3.1.24. Variablenübergabe und -verarbeitung

Durch die Möglichkeit, PHP und die {cp:tags} in beliebigen Bereichen einzeln oder auch in Kombination zu verwenden, entstehen ganz neue Wege der Erstellung von neuen Modulen und Interaktionen.

Um beispielsweise bestimmte Inhalte in Abhängigkeit eines Parameters anzuzeigen (d.h. Parameter wie z.B. "action"), wird dieser beim Aufruf der URL an diese angehängt:

Statt */index.php?id=300 z.B. */index.php?id=300&full_content=yes

Ob dieser Parameter gesetzt ist, können Sie in dem Artikel - bzw. sonst in der Vorlage - mit dem {cp:tag} "{cp:if:full_content}" überprüfen oder Sie setzen PHP ein und schreiben folgendes:

```
<?php
GLOBAL $full_content;
if ($full_content == "yes") {
    // hier der Inhalt, welcher angezeigt werden soll
}
?>
```

Ein Beispiel für eine sinnvolle Verwendung der Variablenübergabe sehen Sie in unserem Lexikon unter http://www.sandoba.de/support/computerlexikon/. Dort ruft ein Artikel (quasi der "Rahmen" des Lexikons) in Abhängigkeit von den übergebenen Variablen die passenden Artikel der Kategorie "Lexikon" auf.

3.2. Häufige Fragen zur Arbeit mit den Vorlagen

3.2.1. Kann ich meine PHP-Skripte weiterverwenden?

Ja, dies ist möglich. Sie haben verschiedene Möglichkeiten in Contentpapst, Ihre Skripte weiterzuverwenden. U.a. können Sie diese über einen Link aus Contentpapst aufrufen, diese in eine Seite inkludieren ("include", einbinden) lassen über den Tag {cp:include} oder diese direkt in einen Artikel oder eine Vorlage integrieren. Wenn Sie PHP-Skripte in Artikeln oder in Vorlagen einbinden, welche für einen Container genutzt werden, wird zusätzlich (optional) die Ausgabe des Skripts zwischengespeichert. Dies führt zu geringeren Ladezeiten.

3.2.2. Können Sie mir eine Entwicklungsumgebung empfehlen?

Wenn Sie Contentpapst um eigene Module / Funktionen erweitern wollen, dann können Sie neue PHP-Dateien prinzipiell in jedem einfachen Texteditor (Notepad, Word) verfassen. Dennoch gibt es speziell für den PHP-Bereich leistungsstarke Editoren, die den Benutzer bei der Arbeit durch Funktionsübersichten, eine farbige Quellcode-Darstellung usw. unterstützen.

Einige besonders empfehlenswerte Editoren sind:

- TextPad - http://www.textpad.com/
- PHP Eclipse - http://phpeclipse.sourceforge.net/
- PHP Edit - http://www.phpedit.net/

- PHP Coder - http://www.phpide.de/

Die hier genannten Entwicklungsumgebungen sind zum Teil kostenlos. Einige sind als Shareware erhältlich und müssen nach der Testphase registriert werden, jedoch ist ein Kauf sehr zu empfehlen (aufgrund der teils erheblichen Zeitersparnis).

3.2.3. Warum können Dateien / Vorlagen nicht bearbeitet werden?

Contentpapst ist bei der Bearbeitung von Dateien immer auf ausreichende Zugriffsrechte angewiesen, d.h. es müssen für die jeweiligen Dateien / Ordner Lese- und Schreibrechte vorhanden sein. Sind diese Rechte nicht verfügbar, dann versucht Contentpapst diese automatisch zu setzen. Je nach genutztem Webspace-Anbieter ist dies allerdings nicht immer möglich. In diesem Fall müssen Sie dann per FTP den CHMOD auf die Dateien / Ordner manuell ändern (auf 777 bzw. 775). Dann kann die Bearbeitung über Contentpapst erfolgen.

3.2.4. Was ist mit Änderungen der PHP-Dateien von Contentpapst?

Sie haben generell die Möglichkeit, Contentpapst-Dateien an Ihre Bedürfnisse anzupassen. Da Contentpapst in PHP geschrieben ist, liegt der Quellcode für Veränderungen offen. Änderungen an diesen Dateien unterliegen jedoch den Lizenzbestimmungen. Es kann keine Gewährleistung auf die Funktionalität des Systems übernommen werden, wenn Contentpapst-Dateien von Ihnen bearbeitet werden. Änderungen an Funktionen oder bspw. die Erstellung von neuen Modulen nach Ihren Wünschen können auch direkt durch die sandoba.de medien agentur vorgenommen werden. Sie profitieren doppelt: durch das Wissen der Entwickler und die gesparte Entwicklungszeit. Kontaktieren Sie uns und wir erstellen Ihnen ein unverbindliches Angebot.

4. Einsatz der {cp:tags}

Die {cp:tags} werden zum Einbau dynamischer Seitenbestandteile in die Vorlagen genutzt. Mit Hilfe dieser Tags lassen sich beispielsweise einzelne Seiten mit einer Auflistung der News oder mit verschiedenen Werbemitteln versehen. Auch die Navigation kann in ihrer Art und Position unterschiedlich angeordnet werden. Die Möglichkeiten sind vielfältig. Alle im System nutzbaren {cp:tags} werden in dieser Kategorie kurz erläutert, {cp:tags} und Container für bestimmte Module werden dagegen direkt in der entsprechenden Kategorie dieser Anleitung unterhalb von "Systemkomponenten / Module" beschrieben.

4.1. Übersicht

4.1.0. Allgemeines zu den {cp:tags}

Die {cp:tags} sind eine in **Contentpapst** integrierte Metasprache, durch welche sich bestimmte Elemente innerhalb der Vorlagen oder auch in Artikel einfach positionieren lassen. Dazu gehören die Anzeige der neuesten Links, eine Möglichkeit zum Wechsel der Sprache usw. Die Syntax beginnt dabei immer mit "cp:" und wird in geschweifte Klammern eingefasst. Einige {cp:tags} müssen zusätzlich geschlossen werden.

Die Syntax baut auf folgender Schreibweise auf (am Beispiel von "{cp:container}"):

```
{cp:container module="news" start="0" limit="5" [order=""] [dir=""] [category=""]}
```

Der Name des Tags ist hierbei "container". Alle Angaben rechts davon innerhalb des Tags sind Parameter, welche z.B. die Art der Daten oder die Darstellung dieser Daten beeinflussen. Die Attribute dieser Parameter sind generell in Hochkommata eingeschlossen, Leerzeichen sind nur zwischen den Parametern zulässig.

Alle Parameter, welche in dieser Anleitung in eckigen Klammern stehen, sind optional und können je nach Bedarf hinzugefügt werden.

4.1.1. Aufbau der Tags

Es wird unterschieden zwischen Tags, welche die weitere Verarbeitung der Vorlagen beeinflussen (bspw. Container, Variablen-Bearbeitung) und Tags, welche lediglich einzelne Inhalte ausgeben. Alle Tags des ersten Typs werden einzeln beschrieben und sind in allen Artikeln/Vorlagen verwendbar, für den zweiten Typ der Tags gibt es einen festgelegten Aufbau. Diese Tags können nur in den Vorlagen des jeweiligen Modules verwendet werden.

Am Beispiel einer Vorlage für eine News (z.B. die Vorlage "news_text.htm" in Ihrem Vorlagen-Set) sehen Sie die Verwendung der Tags. Der Name setzt sich stets zusammen aus "{cp:", dem Namen des Modules sowie dem Namen der Datenbankspalte, in welcher sich der gewünschte Inhalt befindet. Soll Beispielsweise der Titel einer News ausgegeben werden, müssen Sie den Tag "{cp:news:title}" in Ihre Vorlage integrieren. Ein Tag diesen Typs kann mehrfach innerhalb der Vorlage verwendet werden.

4.1.2. Sprachelemente in den Vorlagen: {cpl:tags}

Um Sprachelemente in den Vorlagen nutzen zu können und dennoch diese dynamisch nach der eingestellten Sprache zu ersetzen, wurden die sogenannten {cpl:tags} eingeführt ("**Contentpapst**-Language-Tags"). Im Namen des Tags {cpl:*} wird * durch den Inhalt der entsprechenden Variable aus den Sprachdateien ersetzt, zusammen mit dem Namen des Moduls.

Werden Sprachelemente aus der Hauptdatei (Ordner "/languages/*") genutzt, so ist kein Modul anzugeben. Beinhaltet eine

Vorlage beispielsweise den Tag {cpl:0469} und es ist als Sprache Deutsch ("de") eingestellt, so wird aus dem Ordner "/languages/de/" die Datei "index.php" geöffnet. In dieser wird nach der Variable $lang["0469"] gesucht. Deren Inhalt (in diesem Fall "Pfad zum CMS") ersetzt in der Vorlage dann die ursprünglich vorhandene Angabe "{cpl:0469}".

Bei den Modulen verhält es sich ähnlich. Wenn ein Sprachelement aus dem Modul "CP::Shop" verwendet werden soll, so schreibt man beispielsweise "{cpl:shop:0083}". Hier würde dann die Variable "$lang["shop"]["0083"]" aus der Datei "/modules/shop/languages/de/index.php" angefordert werden. Voraussetzung ist ebenfalls, dass "de" als Sprache voreingestellt ist und das Modul im Ordner "/modules/shop/" abgelegt ist.

4.1.3. {cp:assign}

{cp:assign} dient der Zuweisung von Variableninhalten innerhalb einer Vorlage. Vorhandene Variableninhalte können damit überschrieben werden. In Kombination mit anderen {cp:tags} stellt {cp:assign} eine Möglichkeit dar, z.B. ein Passwort nur verschlüsselt anzuzeigen, wenn dieses vorhanden ist. {cp:assign} erhält zur Verarbeitung den Namen der zu überschreibenden Variablen und deren neuen Inhalt übermittelt.

```
{cp:assign variable="password" value="*****"}
// das Passwort aus der Variable "$password" wird mit "*****" überschrieben
```

4.1.4. {cp:break}

Dieser spezielle Tag dient der Arbeitserleichterung bei der Erstellung von Artikeln. Er steht nicht innerhalb der Vorlagen zur Verfügung. In einem Artikel eingesetzt, erzeugt er einen Seitenumbruch, d.h. der Artikel wird auf mehrere Seiten aufgeteilt. Mittels {cp:pages} kann dann in der genutzten Artikelvorlage eine automatisch generierte Navigation zum Umblättern eingeblendet werden.

4.1.5. {cp:buildtime}

Dieser Tag gibt die Zeit an, welche zum Generieren der aktuellen Seite benötigt wurde. Vor allem in der Test-Phase kann man mit {cp:buildtime} Fortschritte bei der Optimierung der Seiten feststellen. Die einzelnen Seiten werden vom Browser der Reihe nach von oben nach unten generiert. Dementsprechend sollte {cp:buildtime} auch nur am Ende der Seite zur Anzeige der benötigten Zeit genutzt werden.

4.1.6. {cp:configuration}

Ähnlich wie "{cp:variable}" sorgt dieser Tag für die Integration von Systemeinstellungen in die Vorlagen. Kann {cp:variable} lediglich normale String- und Integer-Variablen in die Vorlagen einsetzen, so greift {cp:configuration} direkt auf das Konfigurations-Array zu, in dem alle Einstellungen gespeichert werden. Der Tag baut sich folgendermaßen auf:

```
{cp:configuration:[MODUL]:[EINSTELLUNG]}
```

Anstelle von "[MODUL]" setzen Sie den Namen des gewünschten Moduls ein, bei "[EINSTELLUNG]" entsprechend den Namen der Einstellung. Intern greift **Contentpapst** hier auf "$configuration["[MODUL]"]["[EINSTELLUNG]"] zu.

Welche Namen Sie für die beiden Parameter einsetzen können, erfahren Sie in der Datei "config.db.php" in Ihrem Vorlagen-Set.

4.1.7. {cp:container module="article" type="related"}

Contentpapst bietet dem Vorlagen-Designer die Möglichkeit, innerhalb einer beliebigen Artikel-Vorlage bzw. auch innerhalb eines Artikels verwandte Artikel anzuzeigen. Es werden jeweils Artikel gelistet, deren Schlüsselwörter den Schlüsselwörtern des aktuellen Artikels ähnlich sind.

Hierzu müssen für die gewünschten Artikel jeweils im Artikel-Modul Schlüsselwörter angelegt werden (durch Kommata getrennt). Ist dieser Vorgang abgeschlossen, dann gilt es zu entscheiden, ob die verwandten Artikel innerhalb eines Artikels bzw. innerhalb einer Artikel-Vorlage eingeblendet werden sollen. Wenn bei einer großen Anzahl von Artikeln die Auflistung verwandter Artikel gewünscht ist, dann ist die letztere Variante zu bevorzugen.

Öffnen Sie in diesem Fall bitte eine der Artikel-Vorlagen Ihres Vorlagen-Sets. Bei einem Vorlagen-Set "Business" und einer Artikel-Vorlage "Free" ist dies die Datei "/themes/business/article_free.htm". Suchen Sie sich die gewünschte Stelle für die verwandten Artikel aus und geben Sie folgenden Code ein:

```
{cp:container module="article" type="related" number="{cp:article:number}" limit="5"
template="themes/business/related.htm" return="1"}
```

Durch diesen Code wird ein Container aus dem Artikel-Modul eingebunden. Der Container benötigt zwingend die Angabe der Artikel-ID ("number"). Optional sind die Angabe von "limit" (max. Anzahl der verwandten Artikel) und "template" (verwendete Vorlage). Wird keine Vorlage angegeben, dann wird die Standard-Vorlage "articles_related.htm" aus dem aktuellen Vorlagen-Set verwendet. Alle Angaben können frei variiert werden. Für "number" ist selbstverständlich auch direkt die Angabe der Artikel-ID möglich.

Der Container ruft jetzt für jeden verwandten Artikel diese Vorlage auf und wird anhand der dort vorgenommenen Formatierungen ausgegeben. Eine Ausgabe in Form einer Liste mit dem Titel der Artikel könnte demnach folgendermaßen ausschauen:

```
- <a href="{cp:article:link}">{cp:article:header}</a><br />
```

Wenn zu den Schlüsselwörtern eines Artikels keine verwandten Artikel existieren, dann wird auch kein Ergebnis zurückgeliefert. Über den optionalen Parameter "return" ("... return="1" ...") kann jedoch auch eine entsprechende Mitteilung geliefert werden.

4.1.8. {cp:container module="article"}

Mit {cp:container module="article"} ist die freie Positionierung von Artikelinhalten auf der Webseite möglich, unabhängig von der standardmäßigen Ansicht aus "index.php?file=article". Der Tag {cp:container module="article"} besteht im Wesentlichen aus einer Reihe von Angaben, welche Ordnung, Art und Umfang der Ausgabe beeinflussen. {cp:container module="article"} enthält Angaben zu Kategorie, Anzahl/Begrenzung, Ordnung, sowie der Ordnungsrichtung:

start	Artikel ab diesem Eintrag ausgeben
limit	Anzahl der Artikel
category	Artikel aus einer bestimmten Kategorie (bei allen Kategorien "-" einsetzen)
subcategories	Mit "subcategories = "yes" " werden auch Unterkategorien gezeigt
order	Artikel ordnen (möglich sind "date", "number", "title" und "random")
dir	Aufsteigend ("asc") oder absteigend ("desc") ordnen
length	Anzahl der Zeichen des Artikeltextes
template	Pfad zu einer Vorlage (andernfalls wird Standardvorlage genutzt)
format	Format der Ausgabe (Text/HTML)
cache	Zeit der Zwischenspeicherung in Sekunden

Wenn Sie eine Begrenzung nach einer Kategorie vornehmen möchten, erhalten Sie die entsprechende Kategorienummer unter "Artikel > Kategorien".

Anwendungsbeispiele:

```
{cp:container module="article" start="1" limit="1" category="-" order="date" dir="desc" length="500"}
```

Gibt die ersten 500 Zeichen des zuletzt auf der Webseite veröffentlichten Artikels aus. Es wird nicht nach Kategorien unterschieden.

```
{cp:container module="article" start="1" limit="5" category="1" order="title" dir="asc" length="200"}
```

Gibt die ersten fünf Artikel aus der Kategorie "1" aus. Geordnet wird alphabetisch in aufsteigender Reihenfolge. Es werden jeweils 200 Zeichen des Artikels ausgegeben.

Einsatz:

Der Tag {cp:container module="article"} wird vor allem innerhalb der Vorlagen für die Startseiten sowie für die Hauptseite eingesetzt. Jedoch kann dieser auch innerhalb von Artikeln genutzt werden. Denkbar wäre demnach z.B. das Anlegen einer neuen Artikel-Vorlage, welche dann mit Inhalten aus einem Artikel gefüllt wird, um beispielsweise eine getrennte Artikel-Übersicht zu generieren.

4.1.9. {cp:container module="fast_gallery"}

Neben dem standardmäßig in **Contentpapst** integrierten Galerie-Modul wurde das Modul "fast_gallery" zur unkomplizierten Anzeige von Grafiken in einem Verzeichnis entwickelt. "fast_gallery" wird über Container direkt in eine beliebige Vorlage eingebettet. Dies ermöglicht beispielsweise die Anzeige einer Galerie mit themenbezogenen Bildern direkt in einem Artikel, ohne dass alle Bilder einzeln hochgeladen und eingefügt werden müssen. Ebenso nimmt einem die Galerie Arbeiten wie die Größenanpassung und Optimierung der Grafiken usw. ab.

Folgende Angaben sind zulässig / erforderlich:

path	Der Pfad zum anzuzeigenden Verzeichnis
start	Anzeige ab Bild "n" (optional)
width	Breite der Grafiken (optional)
height	Höhe der Grafiken (optional)
cols	Spaltenanzahl der Tabelle
order	Grafiken sortieren nach ("random" = zufällig)
rows	Zeilenanzahl der Tabelle
type	Zugelassene Dateiformate (komma-separiert, optional)
thumbnails	Vorschau-Bilder generieren (0 = nein, 1 = ja)
popup	Popup auf Klick (0 = nein, 1 = ja, 2 = Popup per Lightbox)

Zusammenfassung (mit Beispiel):

```
{cp:container module="fast_gallery" path="__VERZEICHNIS__" [start="0"] [width="200"] [height="100"] cols="3"
rows="3" [type="jpg,gif"] [popup="1"] [thumbnails="1"]}
```

Anwendungsbeispiele:

```
{cp:container module="fast_gallery" path="images/reisen" start="0" width="150" cols="2" rows="4"
type="png,jpg,swf" popup="1"}
```

Zeigt insgesamt 8 Bilder (2 Spalten à 4 Zeilen) aus dem Verzeichnis "images/reisen" an, angefangen vom ersten Bild. Gelistet werden nur Bilder in den Formaten "png", "jpg" und "swf". Die Bilder werden auf eine Breite von 150 Pixel skaliert, zwischengespeichert und mit einem Link zum Popup versehen.

Einsatz:

Der Tag {cp:container module="fast_gallery"} kann für jegliche Arten von Galerien / Bildübersichten genutzt werden, für die die Funktionen des normalen Galerie-Moduls zu umfangreich sind. Denkbar wäre z.B. eine Übersicht mit Produktscreenshots zu einer Werbepräsentation, deren Vollansicht erst nach einem Klick auf eines der Vorschau-Bilder sichtbar wird.

4.1.10. {cp:container module="friend"}

Im Unterschied zu {cp:friend}, kann mit diesem Modul ein komplettes Formular zur Weiterempfehlung der Webseite auf einer beliebigen Seite eingeblendet werden. Es werden Felder für den eigenen Namen sowie die Email-Adresse, Name/Email-Adresse des Freundes und ein Feld für die eigentliche Mitteilung angezeigt.

Durch den Containeraufruf können bereits einige Standard-Werte vorgegeben werden, welche dann beim Aufruf des Formulares angezeigt werden. Der Containeraufruf wird folgendermaßen angewendet:

```
{cp:container module="friend" from_name="ABSENDERNAME" from_email="ABSENDEREMAIL"
to_name="EMPFÄNGERNAME" to_email="EMPFÄNGEREMAIL" content="TEXT" template="vorlage.htm"
return="ergebnisseite.htm"}
```

Jede dieser Parameter im Containeraufruf ist optional (bis auf "module"}. Wenn keine Vorlage angegeben ist, dann wird zum Aufbau des Formulares die Vorlage "friend.htm" im aktuellen Vorlagen-Set verwendet. Ohne Angabe des Parameters "return" wird nach dem Versand der Email auf eine Seite mit einer Standardmeldung zurückgesprungen.

4.1.11. {cp:container module="links"}

Einer der leistungsfähigsten Tags von **Contentpapst**. Er ermöglicht den freien Einsatz von Links aus der Datenbank auf der Webseite, unabhängig von der standardmäßigen Link-Ansicht aus "index.php?file=links". Der Tag {cp:container module="links"} besteht im Wesentlichen aus einer Reihe von Angaben, welche die Ordnung und Art der Ausgabe beeinflussen. {cp:container module="links"} enthält Angaben zu Kategorie, Anzahl/Begrenzung, Ordnung, sowie der Ordnungsrichtung:

start	Links ab diesem Eintrag ausgeben
limit	Anzahl der Links
category	Links aus einer bestimmten Kategorie (bei allen Kategorien "-" einsetzen)
order	Links ordnen (möglich sind "date", "hits", "number" und "title")
dir	Aufsteigend ("asc") oder absteigend ("desc") ordnen
template	Pfad zu einer Vorlage (andernfalls wird Standardvorlage genutzt)

Wenn Sie eine Begrenzung nach einer Kategorie vornehmen möchten, erhalten Sie die entsprechende Kategorienummer unter "Module > Links > Kategorie-Übersicht".

Anwendungsbeispiele:

```
{cp:container module="links" start="1" limit="1" category="-" order="date" dir="desc"}
```

Gibt den zuletzt auf der Webseite veröffentlichten Link aus. Es wird nicht nach Kategorien unterschieden.

```
{cp:container module="links" start="1" limit="5" category="1" order="title" dir="asc"}
```

Gibt die ersten fünf Links aus der Kategorie "1" aus. Geordnet wird alphabetisch in aufsteigender Reihenfolge.

Einsatz:

Der Tag {cp:container module="links"} wird vor allem innerhalb der Vorlagen für die Startseiten sowie für die Hauptseite eingesetzt. Jedoch kann dieser auch innerhalb von Artikeln genutzt werden. Denkbar wäre demnach z.B. das Anlegen einer neuen Artikel-Vorlage, welche dann mit Inhalten aus einem Artikel gefüllt wird, um beispielsweise eine Link-Übersicht zu generieren.

4.1.12. {cp:container module="news" type="list"}

{cp:container module="news" type="list"} zeigt die Newskategorien zusammen mit einem Link zur Seite "News einschicken" an. Mit den Einstellungen unter "News > Modul-Einstellungen" können zusätzlich die Anzahl der News, die Anzeige der Kategoriebeschreibung sowie verschiedene Benachrichtigungsregeln beeinflusst werden.

4.1.13. {cp:container module="news"}

Einer der leistungsfähigsten Tags von **Contentpapst**. Er ermöglicht den freien Einsatz von News aus der Datenbank auf der Webseite, unabhängig von der standardmäßigen News-Ansicht aus "index.php?file=news" (bzw. "news.html"). Der Tag {cp:container module="news"} besteht im Wesentlichen aus einer Reihe von Angaben, welche die Ordnung und Art der Ausgabe beeinflussen. {cp:container module="news"} enthält Angaben zu Kategorie, Anzahl/Begrenzung, Ordnung, sowie der Ordnungsrichtung:

start	News ab diesem Eintrag ausgeben
limit	Anzahl der News
category	News aus einer bestimmten Kategorie (bei allen Kategorien "-" einsetzen)
order	News ordnen (möglich sind "date", "number" und "title")
dir	Aufsteigend ("asc") oder absteigend ("desc") ordnen

Wenn Sie eine Begrenzung nach einer Kategorie vornehmen möchten, erhalten Sie die entsprechende Kategorienummer unter "News-System > Kategorie-Übersicht".

Anwendungsbeispiele:

```
{cp:container module="news" start="1" limit="1" category="-" order="date" dir="desc"}
```

Gibt die zuletzt auf der Webseite veröffentlichte News aus. Es wird nicht nach Kategorien unterschieden.

```
{cp:container module="news" start="1" limit="5" category="1" order="title" dir="asc"}
```

Gibt die ersten fünf News aus der Kategorie "1" aus. Geordnet wird alphabetisch in aufsteigender Reihenfolge.

Einsatz:

Der Tag {cp:container module="news"} wird vor allem innerhalb der Vorlagen für die Startseiten sowie für die Hauptseite eingesetzt. Jedoch kann dieser auch innerhalb von Artikeln genutzt werden. Denkbar wäre demnach z.B. das Anlegen einer neuen Artikel-Vorlage, welche dann mit Inhalten aus einem Artikel gefüllt wird, um beispielsweise eine News-Übersicht zu generieren.

4.1.14. {cp:container module="newsletter"}

Mit diesem Containeraufruf lässt sich auf der Webseite ein Block zur Newsletteranmeldung einblenden. Es sind dabei immer nur die unter "Newsletter > Modul-Einstellungen" freigeschalteten Felder sichtbar. Ein Auswahlfeld für den Newsletter wird erst eingeblendet, sobald mehr als ein Newsletter vorhanden und aktiviert ist.

4.1.15. {cp:container module="search"}

Mit {cp:container module="search"} lässt sich ein Container für die Volltextsuche an einer beliebigen Stelle der Webseite platzieren. Über einen optionalen Parameter "template" kann angegeben werden, welche Vorlage für den Container verwendet werden soll. Ist dieser Parameter nicht vorhanden, wird automatisch auf die Vorlage "/modules/search/templates/search_form.htm" zurückgegriffen.

Dies ermöglicht die Darstellung verschiedener Suchcontainer je nach Bereich der Webseite. Das Suchformular kann zusätzlich über ein Feld "category" den Suchbereich beeinflussen. Möglich sind hier standardmäßig "news" und "articles".

Anwendungsbeispiele:

```
{cp:container module="search" template="suche_artikel.htm"}
```

Bindet den Suchcontainer mit der Vorlage "suche_artikel.htm" ein. Der Bereich der Suche ist in dieser Vorlage automatisch ausgewählt über ein verstecktes Formular-Feld.

```
{cp:container module="search"}
```

Bindet den Suchcontainer mit der Standard-Vorlagenset (siehe oben) auf der Webseite ein.

4.1.16. {cp:container module="statistic"}

In einigen Fällen kann es angebracht sein, Besucher auch öffentlich über die Statistik der Webseite zu informieren. {cp:container module="statistic"} kann hierzu in das Layout einer beliebigen Vorlage integriert werden. Der Tag liefert Angaben über die aktuellen Besucher sowie die Hits der Seite heute, gestern und insgesamt.

Optional kann man sich über die Angabe "visitors" (z.B. "{cp:container module="statistic" visitors="today"}") auch nur bestimmte Angaben über die Statistik zurückliefern lassen. Mögliche Attribute sind hierbei "today" (Seitenaufrufe heute), "yesterday" (Seitenaufrufe gestern), "total" (Seitenaufrufe insgesamt) und "online" (Anzahl der aktuellen Besucher).

4.1.17. {cp:date}

Mit {cp:date} können Sie ein formatiertes Datum auf Ihrer Webseite anzeigen lassen. Im Gegensatz zu {cp:variable:date} und {cp:variable:dateshort} wird hier kein festgelegtes Format genutzt. Vielmehr wird über den Parameter "format" das

gewünschte Format des Datums angegeben. Genauere Informationen zu den möglichen Zeichenketten erhalten Sie unter "http://www.php.net/date". Als zweiter Parameter kann mit "time" ein Datum im Timestamp-Format angegeben werden. Ist dieser Parameter nicht gesetzt, wird das aktuelle Datum verwendet.

```
{cp:date format="d.m.Y"} // liefert Datumsangabe, z.B. 23.05.2003
{cp:date format="H:i" time="1075754007"} // liefert Zeitangabe, z.B. 18:32 (hier für den Timestamp "1075754007")
```

4.1.18. {cp:if}

Mittels des Konstrukts {cp:if} lässt sich das Vorhandensein einer Variablen direkt in einer Vorlage überprüfen. Dies können Variablen aus dem Modul oder auch aus den Konfigurationseinstellungen sein. Variablen sind in PHP durch ein "$" am Anfang des Variablennamens gekennzeichnet (z.B. "$symbol"). Ein Beispiel soll den Sinn von {cp:if} verdeutlichen:

```
{cp:if:email}
<table border="0"><tr bgcolor="#A32333"><td><font color="#FFFFFF">Unser Support: {cp:variable:email}
</font></td></tr></table>
{/cp:if:email}
```

In diesem Beispiel wird geprüft, ob die Variable "$email" vorhanden und mit Inhalt gefüllt ist. Ist dies der Fall, dann wird eine Tabelle eingeblendet mit innenliegendem Text und der Email-Adresse. Diese wird dann über {cp:variable} eingeblendet.

4.1.19. {cp:ifnot}

Mittels des Konstrukts {cp:ifnot} lässt sich überprüfen, ob eine Variable NICHT vorhanden ist. Dies können Variablen aus dem Modul oder auch aus den Konfigurationseinstellungen sein. Variablen sind in PHP durch ein "$" am Anfang des Variablennamens gekennzeichnet (z.B. "$symbol"). Ein Beispiel soll den Sinn von {cp:ifnot} verdeutlichen:

```
{cp:ifnot:username}
<table border="0"><tr bgcolor="#A32333"><td><font color="#FFFFFF">Sie Sind nicht eingeloggt, bitte <a
href="index.php?file=user">hier</a> einloggen.</font></td></tr></table>
{/cp:ifnot:username}
```

In diesem Beispiel wird geprüft, ob die Variable "$username" NICHT vorhanden ist. Ist dies der Fall, dann wird eine Tabelle eingeblendet mit innenliegenden Text und einem Link zur Login-Seite.

4.1.20. {cp:ignore} ... {/cp:ignore}

Wenn Sie in Artikeln oder Vorlagen {cp:tags} einsetzen wollen, welche nicht verarbeitet werden sollen, so können Sie dies dem Parser mitteilen, indem Sie die betroffenen Bereiche mit {cp:ignore} ... {/cp:ignore} umgeben. Dies ist z.B. bei Anleitungen zu den {cp:tags} nützlich, wie diese unter http://www.sandoba.de/ in der Knowledge Base zu finden sind.

4.1.21. {cp:include}

Contentpapst stellt Ihnen eine Vielzahl an Funktionen für Ihre Webseite bereit. Benötigen Sie jedoch eine spezielle Funktion und wollen nicht bis zur nächsten Version warten, so können Sie auch andere, bereits vorhandene Scripte einbinden. Hierfür wurde mit {cp:include} ein eigener Tag angelegt, welcher einfach in den Vorlagen eingesetzt wird.

Zur Einbindung eines Scripts mit dem Namen "weblog.php" ist Folgendes in die Vorlage einzubauen:

```
{cp:include}weblog.php{/cp:include}
```

Wenn das Script in einem anderen Ordner liegt, so geben Sie zusätzlich den Pfad zu dieser Datei ausgehend vom Hauptordner an. Für PHP-Dateien sollten Sie zusätzlich den Parameter "type" mit dem Wert "php" anhängen ("... type="php" ..."). Dann kann der Inhalt immer korrekt verarbeitet werden.

4.1.22. {cp:ismember}

Gezielt können in den Vorlagen einzelne Inhalte nur für registrierte und eingeloggte Mitglieder sichtbar gemacht werden. Dazu zählen alle Benutzergruppen des Systems inklusive der Redakteure. Der Tag muss nach den zu schützenden Inhalten wieder geschlossen werden.

```
{cp:ismember}Dieser Text wird nur Mitgliedern gezeigt.{/cp:ismember}
```

4.1.23. {cp:language}

Wenn Ihre Webseite mehrsprachig aufgebaut wird, so müssen Sie Ihren Besuchern die Möglichkeit geben, die Sprache zu wechseln. Dies ist mit dem Tag {cp:language} möglich, durch welchen die Flaggen für die jeweils verfügbaren Sprachversionen eingeblendet werden.

4.1.24. {cp:lastlogin}

Mit diesem Tag können Sie in einer beliebigen Vorlage der Webseite das Datum des letzten Logins eines Benutzers einblenden lassen. Standardmäßig wird dies in der Vorlage des Admininistrations-Bereiches vorgenommen.

4.1.25. {cp:level}

Der Tag {cp:level} zeigt die Position der aufgerufenen Seite innerhalb der Struktur der Webseite per "Backtracking-Navigation" an. Beispielsweise kann dies bei der Galerie oder im Link-Modul genutzt werden und sieht dann folgendermaßen aus: "Galerie > Kategoriename > Bildname". Alle Kategorien sowie die Hauptseite sind mit Links versehen, so dass ein flexibles Wechseln möglich ist.

Auch in den Artikelvorlagen kann der Tag genutzt werden. Hier ist zusätzlich die Erweiterung um den Parameter "base" möglich. Ist "base" nicht gewählt oder "base" = 0, wird der Link zur Artikel-Hauptseite mit angezeigt. Bei "base" = 1 jedoch ist die oberste Ebene die Hauptkategorie des Artikels.

4.1.26. {cp:newslink}

Der Tag {cp:newslink} steht nur in der Vorlage für die News-Detailansicht zur Verfügung. Er beinhaltet jeweils einen der zu der ausgewählten News angegebenen weiterführenden Links mit URL und Beschriftung. Dem Namen des Tags ist die Nummer des Links anzuhängen (1-3). Zwischen öffnendem und schließendem Tag kann eine Zeichenkette integriert werden, welche dem Link vorangestellt wird, sofern dieser vorhanden ist.

```
{cp:newslink1}» {/cp:newslink1} // Ergebnis (Beispiel): » Contentpapst
```

4.1.27. {cp:path}

{cp:path} fügt immer den relativen Pfad zum aktuell genutzten Vorlagen-Set ein. Hiermit können Sie beispielsweise auf Grafiken zugreifen, welche in den Vorlagen-Ordnern liegen, ohne dass Sie den kompletten Pfad mit "themes/NAME_DES_VORLAGENSETS/" stets mit angeben müssen.

4.1.28. {cp:container module="random"}

Mit {cp:container module="random"} integrieren Sie die im Modul "Zufall" angegebenen Inhalte in die Webseite. Hierzu wird mittels des Parameters "type" der Typ bzw. die Kategorie des Inhaltes angegeben. Aus allen entsprechenden Inhalten wird dann jeweils ein Eintrag ausgewählt und auf der Webseite angezeigt.

```
{cp:container module="random" type="1"} // zeigt Inhalte des Typs mit der ID "1" an
```

4.1.29. {cp:rewrite}

Besonders wichtig für die Suchmaschinen-Optimierung der Seite sind URL's, die Schlüsselwörter enthalten, welche auch im Text vorkommen. Mit Hilfe des Tags {cp:rewrite} können derartige URL's leicht erzeugt werden. Über den Parameter "text" wird dem Tag ein kurzer Text bis ca. 50 Zeichen übergeben. Zurückgeliefert wird der Text in einem Format, wie dieses in einer URL verwendet werden kann.

```
{cp:rewrite text="Mit Contentpapst in die Suchmaschinen"} // gibt "mit-contentpapst-in-die-suchmaschinen" zurück
```

4.1.30. {cp:rights}

Der Tag "{cp:rights}" überprüft, ob ein Benutzer über ein bestimmtes Zugriffsrecht verfügt. Wenn dies der Fall sein sollte, dann wird der durch den Tag umschlossene Raum angezeigt. Hierzu muss zusätzlich der Name des Zugriffsrechtes bzw. ein Array mit Namen übergeben werden (letzteres bewirkt eine "UND"-Verknüpfung). "{cp:rights}" kann mit der Angabe "type" auch verschachtelt eingesetzt werden (zwei Ebenen). Eine "OR"-Verknüpfung ist durch das Hinzufügen von "connection="or"" möglich.

Anwendung:

```
{cp:rights name="options"} ... {/cp:rights}
```

Gibt "..." aus, wenn das Zugriffsrecht "options" vorhanden ist.

```
{cp:rights name[0]="article_edit" name[1]="article_delete"}...{/cp:rights}
```

Bei diesem Beispiel wird "..." ausgegeben, wenn sowohl das Zugriffsrecht "article_edit" als auch "article_delete" dem Benutzer zugeteilt sind. Die Namen der möglichen Zugriffsrechte entsprechen dabei dem "Internen Namen", welcher bei einem Zugriffsrecht in der Rechteverwaltung eingegeben wurde.

4.1.31. {cp:systemlogo}

Der Tag {cp:systemlogo} erlaubt die Anzeige des Logos von **Contentpapst**. Dieses ist per Link mit der Webseite http://www.sandoba.de/ verbunden.

4.1.32. {cp:texteffect}

Über den {cp:tag} "{cp:texteffect}" können auf beliebige Texte verschiedene Texteffekte / Formatierungen angewendet werden, ohne dass in den Vorlagen PHP zum Einsatz kommen muss. Diesem Tag muss über den Parameter "type" zuerst der gewünschte Texteffekt übergeben werden. Der Parameter "text" enthält dann den zu formatierenden Text.

Verfügbare Texteffekte / Formatierungen (Parameter "type")	
spacing	Fügt nach jedem Buchstaben ein Leerzeichen ein.

Anwendungsbeispiele:

```
{cp:texteffect type="spacing" text="{cp:variable:title}"}
```

Gibt den Inhalt der Variablen "$title" mit Leerzeichen nach jedem Buchstaben zurück. Bei "Impressum" wäre der Rückgabewert also "I m p r e s s u m".

Einsatz:

Dieser Tag wird seltener bei feststehenden Texten zum Einsatz kommen. Stattdessen kann diesem auch wiederum per {cp:tag} ein dynamischer Text übergeben werden. Eine Einsatzmöglichkeit wäre hier z.B. die Formatierung eines Artikeltitels direkt in der Artikelvorlage, ohne dass der Texteffekt auf jeden Artikel einzeln angewandt werden muss.

4.1.33. {cp:client}

Der {cp:tag} "{cp:client}" stellt eine sinnvolle Möglichkeit dar, gezielt Inhalt nur für bestimmte Browser / Betriebsysteme oder Browser mit aktiviertem / deaktiviertem Javascript auszuliefern. Dies ist z.B. nützlich, wenn für einen Browser andere CSS-Stylesheets verwendet werden sollen.

Dieser Tag umschließt die Informationen, welche nicht für jeden zugänglich sein sollen. Über die Parameter "browser", "os" und "javascript" wird dann gesteuert, wem genau die Informationen angezeigt werden. Dabei können die Parameter beliebig miteinander kombiniert werden:

```
{cp:client browser="msie" os="win" javascript="1"} ... {/cp:client}
// zeigt den Inhalt nur unter Windows mit dem Internet Explorer und aktiviertem Javascript an
```

Werte für den Parameter "browser" (Internet-Browser)	
msie	Internet Explorer
opera	Opera
icab	iCab
lynx	Lynx
konquerer	Konquerer
spider / bot	Suchmaschinen
Netscape	Netscape
Firebird / Firefox	Firefox
mozilla	Mozilla
phoenix	Phoenix
safari	Safari
hotjava	HotJava

Werte für den Parameter "os" ("Operating System" / Betriebssystem)	
win	Windows
linux	Linux
mac	Macintosh
freebsd	FreeBSD
sunos / i86	SunOS
bot / spider	Suchmaschinen
unix / hp-ux / irix / X11	Unix
aix	AIX

Werte für den Parameter "javascript" (Javascript)	
0 / 1	Javascript ist im Browser aktiviert / deaktiviert

4.1.34. {cp:container module="user"}

Dieser Tag integriert ein Formular für das Einloggen im handlichen Format in die Seite. Ist der Benutzer bereits eingeloggt, wird eine Liste mit Links zur Mitgliederliste, zur Bearbeitung des Profiles sowie zur Abmeldung eingeblendet. Genutzt werden kann der Tag in den allgemeinen Seitenvorlagen.

4.1.35. {cp:variable}

Über "{cp:variable:*}" können Sie auf die aktuelle PHP-/Server-Konfiguration zugreifen und diese z.B. in den Artikeln und den Vorlagen weiterverwenden (siehe dazu auch "Einstellungen > Server" im Administrationsbereich). Ebenfalls kann über "{cp:variable:*}" auf Parameter aus der URL zugegriffen werden. Wichtig ist der {cp:tag} "{cp:variable:language}", der stets die aktuelle Sprache des Benutzers enthält, während die standardmäßige Sprache der Webseite über "{cp:configuration:main:language}" abgerufen werden kann.

Beispiele für die Verwendung dieses {cp:tags}	
{cp:variable:language}	genutzte Sprache
{cp:variable:REQUEST_URI}	"$REQUEST-URI"

5. Systemkomponenten / Module

*Module sind eigenständige Teile des Systems. Dies sind beispielsweise Kalender, ein Online-Shop, Foren oder Galerien. Module greifen u.a. auf zentrale Dateien von **Contentpapst**, die Entwickler-API, sowie auf die jeweils genutzten Templates und Sprachdateien zu. Aufgrund dieser Modularisierung lässt sich das System schnell und einfach erweitern. Bei der Grundversion nicht enthaltene Module können auch später noch eingebaut werden.*

5.1. Übersicht

5.1.0. Allgemeines zu den Modulen

Module sind eigenständige Teile des Systems. Dies sind beispielsweise Kalender, Foren oder Galerien. Module greifen u.a. auf die Konfigurations-Datei des Systems, sowie auf die jeweils genutzten Templates, die Rechteverwaltung und Sprachdateien zu. Aufgrund dieser Modularisierung lässt sich **Contentpapst** schnell und einfach erweitern. Weitere, bei der Grundversion nicht enthaltene Module, können auf Wunsch auch später eingebaut werden. Informieren Sie sich dazu unter http://www.sandoba.de/ oder schicken Sie direkt eine Anfrage an "contact@sandoba.de".

5.1.1. Aufruf der Module

Fast alle Module innerhalb des **Contentpapst**-Frameworks können über die URL "index.php?file=[MODULNAME]" für den öffentlichen Bereich bzw. "admin.php?file=[MODULNAME]" für den Administrationsbereich aufgerufen werden. Voraussetzung hierfür ist, dass die entsprechenden Dateien vorhanden sind und das Modul als Unterordner im Ordner "modules" liegt. Anstelle von "[MODULNAME]" ist der Ordnername des Modules einzusetzen. CP::Forms-Module können im Gegensatz dazu jedoch nicht über die URL "index.php?file=[MODULNAME]" aufgerufen werden. Diese werden stattdessen z.B. innerhalb der Vorlagen über spezielle Container aufgerufen (weitere Informationen dazu erhalten Sie unter Systemkomponenten / Module > CP::Forms").

Viele Module bieten darüberhinaus eigene {cp:tags} für die Inhalte. Die Inhalte des Modules müssen also nicht unbedingt nur über die direkte URL aufgerufen werden, sondern es ist auch eine Verwendung auf anderen Seiten innerhalb des Systems möglich.

5.2. Artikel

*Artikel sind eine der einfachsten Möglichkeiten in **Contentpapst**, frei strukturierte Inhalte zu erstellen. Im Gegensatz zu den News sind Sie hier nicht an bestimmte Vorgaben gebunden, sondern Sie können in verschiedenen Kategorien mittels eines WYSIWYG-Editors Texte frei formatieren und bearbeiten.*

5.2.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=article	Aufruf über den Administrationsbereich
index.php?id=[ID]	Aufruf eines Artikels über die ID

5.2.2. Artikel löschen

Artikel, welche nicht mehr auf der Webseite erscheinen und auch nicht im Administrationsbereich zur Verfügung stehen sollen, können in der Artikelübersicht durch einen Klick auf gelöscht werden. Gelöschte Artikel lassen sich im Notfall durch vorherige Backups wiederherstellen. Die Aufspielung dieser Daten sollte jedoch nur von einem Administrator mit entsprechenden Kenntnissen durchgeführt werden.

5.2.3. Artikelvorlagen

Hier handelt es sich nicht um eine spezielle Vorlage, sondern vielmehr um eine variable Anzahl von Vorlagen, welche für

die Erstellung eines Artikels genutzt werden können. Dateien mit dem Dateinamen "include_*.htm" (wobei für "*" ein beliebiger Begriff zu wählen ist) im jeweiligen Vorlagenordner werden automatisch für die Artikel, aber auch für HTML-Dateien in der Dateiverwaltung bereitgestellt.

So können z.B. diverse Vorlagen für einzelne Kategorien angelegt oder Textbausteine gespeichert werden. Soll ein derartiger Baustein genutzt werden, so kann dieser mittels einer Liste schnell auf einen Klick hinzugefügt werden.

5.2.4. Bearbeitung von Artikeln über die Webseite



Durch das sogenannte "Frontend Editing" ist es für die Redakteure in Contentpapst möglich, Artikel direkt über die Webseite zu bearbeiten, ohne dass vorher in den Administrationsbereich gewechselt und der entsprechende Artikel (über die ID bzw. die Suchfunktion) herausgesucht werden muss. Eingeloggten und für die Bearbeitung der Artikel berechtigten Benutzern präsentieren sich die Artikel mit einem zusätzlichen Menü in der linken oberen Ecke der Seite.

Über die Menüpunkte in diesem aufklappbaren Menü kann dann mit einem Klick der Artikel in einem WYSIWYG- oder Text-Editor geladen werden. Nach der Bearbeitung wird der Editor abgeschickt und sofort steht der Artikel online für die Besucher in der neuen Version zur Verfügung.

Ein weiterer Menüpunkt erlaubt den Wechsel in den Administrationsbereich, direkt auf das ausführliche Formular zum Bearbeiten des Artikels. Sollten einmal außer dem Inhalt weitere Angaben zum Artikel geändert werden müssen, dann kann einfach über diesen Menüpunkt gewechselt werden.

Der letzte Menüpunkt ermöglicht ein Löschen des Artikels. Dieser Menüpunkt ist mit einer zusätzlichen Sicherheitsabfrage versehen und erscheint nur bei Benutzern, die auch über die Berechtigung zum Löschen von Artikeln verfügen. Damit wird ein versehentliches Löschen verhindert.

5.2.5. Bewertungen

Artikel können von den Besuchern/Mitgliedern bewertet werden. Nutzen Sie für die Bewertungen die Tags {cp:vote} und {cp:voteset}. Optimal ist eine Kombination mit {cp:ismember} oder {cp:isvisitor}, um die Möglichkeiten zur Bewertung einzuschränken. Die abgegebenen Stimmen werden in der Datenbank gespeichert und mittels {cp:rating} wieder im Artikel ausgegeben.

5.2.6. Direkter Aufruf eines Artikels per URL

Die Artikel können jeweils über verschiedene Wege per URL aufgerufen werden. Dies sind die dynamische URL mit der ID eines Artikels sowie eine Suchmaschinenfreundliche URL samt ID und Titel. Für letztere müssen die Suchmaschinenfreundlichen URL's in den Einstellungen aktiviert werden:

```
*/index.php?id=124 // URL mit ID des Artikels
*/article-124-titel-des-artikels.html // Aufruf des Artikels als simulierte HTML-Datei
```

5.2.7. Kategorien

Artikel können zu Haupt- und Unterkategorien zugeordnet werden. Jede dieser Kategorien besteht aus Titel, Beschreibung und eventuell einem Link zu einem Artikel (mittels ID).

Auf der Startseite der Datei "index.php?file=article" besteht eine Auflistung der aktiven Hauptkategorien mit allen zugeordneten Artikeln. Kategorien können allerdings auch separat aufgerufen werden unter "index.php?file=article&mode=section&number=KATEGORIE-ID". Ist für die Kategorie eine Artikel-ID angegeben, so wird dieser beim Aufruf der Kategorieansicht angezeigt. Andernfalls werden alle Unterkategorien samt Artikeln eingeblendet.

5.2.8. Sichtbarkeit im öffentlichen Bereich

Im Artikel-Modul können Sie für jeden Artikel die Sichtbarkeit im öffentlichen Bereich nach Gruppen einschränken. Dies ist nicht mit der detaillierten Rechteverwaltung für Artikel im Administrations-Bereich (Artikel hinzufügen, bearbeiten etc.) zu verwechseln.

Über die Einschränkung der Sichtbarkeit ist die Bereitstellung von Informationen für einzelne Gruppen wie bspw. Geschäftspartner möglich (z.B. aktuelle Produktdownloads), ohne dass andere Benutzer darauf Zugriff haben. Die Sichtbarkeit wird für jeden Artikel direkt in der Bearbeitungsansicht festgelegt.

Intern regelt Contentpapst dann den Zugriff auf diese Artikel und leitet automatisch auf ein Login-Formular um, wenn die momentanen Rechte des Benutzers nicht ausreichend sind. Nach einem erfolgreichen Login wird wieder zurückgewechselt zum nun sichtbaren Artikel. Die PDF- und Print-Version sowie die Einblendung von Artikeln über einen Container wird von Contentpapst ebenfalls überwacht.

5.2.9. Webservices / Export als RSS-Feed

Informationen zu neu eingestellten / geänderten Artikeln können auch als RSS-Feeds bereitgestellt werden. Dies ermöglicht es anderen Benutzern z.B. diese Informationen auf der eigenen Webseite einzubinden oder über einen Feed-Reader immer auf dem aktuellsten Stand zu bleiben. Es werden jeweils nur freigeschaltete Artikel exportiert, die innerhalb der Laufzeit liegen.

URL's für den Export der Einträge	
webservices.php?file=articl...	RSS-Format (Version 2.0)

Über die folgenden Parameter kann die Auswahl der Artikel weiter gefiltert werden:

Optionale Parameter für den Export	
limit	Maximale Anzahl der Artikel (standardmäßig 10 Artikel)
start	Anzeige ab Artikel "n"
section	ID einer Artikel-Kategorie

5.2.10. Webservices / Export im XML-Format

Das Contentpapst-Framework ermöglicht mittels Webservices u.a. den Zugriff auf die Artikeldaten. Dies ist wichtig, wenn Sie die Inhalte über Contentpapst pflegen, diese allerdings dann z.B. in Flash-Filme integrieren oder in anderen Systemen weiterverwenden wollen.

Es wird darauf geachtet, dass nur freigeschaltete Artikel ausgegeben werden, die innerhalb der Laufzeit liegen. Für die Abfrage eines einzelnen Artikels müssen Sie lediglich die untenstehende URL aufrufen und um die ID/Nummer des Artikels ergänzen (im Artikel-Modul sichtbar). Optional kann ein Parameter "format" angehängt werden, dessen Wert "text" oder "html" sein kann. Standardmäßig steht "format" auf "text", d.h. alle HTML-Tags werden entfernt:

```
webservices.php?file=article&number=302&format=text
// zeigt den Artikel mit der ID "302" im Text-Format an (sofern verfügbar)
```

5.2.11. WYSIWYG-Editor

Der interne WYSIWYG-Editor bietet Desktop-Feeling durch viele Funktionen, welche von Programmen aus dem Desktop-Bereich bekannt sind. Im Folgenden finden Sie die aktuell unterstützen Funktionen noch einmal aufgelistet:

Fett schreiben	Linksbündig	Rückgängig
Kursiv schreiben	Zentriert	Wiederherstellen
Unterstreichen	Rechtsbündig	Drucken
Durchstreichen	Einzug verkleinern	Einzug vergrößern
Nummerierung	Textfarbe	Als separate Datei speichern
Aufzählung	Hintergrundfarbe	Blocksatz
Ausschneiden	Kopieren	Einfügen
Link hinzufügen	Link entfernen	Horizontale Linie
Sonderzeichen einfügen	Formatierung entfernen	

Bei der Auswahl von Textfarbe, Hintergrundfarbe und Grafik öffnet sich jeweils ein neues Fenster, in welchem man durch einen einfachen Klick und eventuell durch einen Klick auf "Abschicken" das gewünschte Element in den WYSIWYG-Editor übernehmen kann. Nachfolgend eine Reihe von erweiterten Funktionen:

- Modus wechseln (wechselt zwischen WYSIWYG- und HTML-Ansicht)
- Quellcode aufräumen (entfernt Tags aus Word und Co.)
- Bilder einfügen (mittels Medienbibliothek vom Webspace)
- Vorschau anzeigen (öffnet ein neues Browser-Fenster)

Tastatur-Shortcuts:

Verschiedene Funktionen des WYSIWYG-Editors in Contentpapst lassen sich auch direkt per Tastatur-Shortcuts ansprechen. Die wichtigsten dieser Shortcuts sind nachfolgend aufgelistet:

Ctrl + A	Alles auswählen	Ctrl + I	Kursiv
Ctrl + C	Kopieren	Ctrl + U	Unterstrichen
Ctrl + X	Ausschneiden	Ctrl + P	Druckdialog
Ctrl + V	Einfügen	Ctrl + M	Absatz erzeugen
Ctrl + F	Suchfeld	Ctrl + Z	Rückgängig
Ctrl + B	Fettschrift	Ctrl + Y	Wiederherstellen

Alternativ zum internen WYSIWYG-Editor können unter "Einstellungen > Allgemeine Einstellungen" als WYISWYG-Editor auch der "FCKeditor" sowie der "tinyMCE"-Editor aktiviert werden.

5.2.12. Zugriffssperre

Um sicherzustellen, dass Arbeiten an Artikeln nicht behindert werden, wird der Zugriff auf die Artikel in der Datenbank gespeichert. Greift ein zweiter Redakteur auf einen bereits durch einen anderen Redakteur geöffneten Artikel zu, so wird diesem der Zugriff verwehrt. Ebenso wird bei der Löschung von Artikeln verfahren.

Der Zugriff auf den Artikel wird entweder durch die Speicherung wieder freigegeben oder durch den Ablauf einer festgelegten Zeitspanne. Bei Contentpapst sind dies standardmäßig 30 min. D.h. wenn nach dieser Zeit noch keine Speicherung erfolgt ist, wird der Artikel dennoch wieder zur Bearbeitung für andere Redakteure freigegeben. Durch diese Regelung kann die Redaktionsarbeit ohne Behinderungen und bei gleichzeitiger Sicherung der Artikel erfolgen. Sollte eine schnellere Freigabe nötig sein, haben Sie immer die Möglichkeit, mit dem zuständigen Redakteur Kontakt aufzunehmen.

5.3. Benutzerverwaltung

Die Benutzer- / Rechteverwaltung ermöglicht sowohl die Bearbeitung der Benutzer, als auch das Versenden von Emails an einzelne Benutzer. Wenn Sie auf Ihrer Webseite eine besondere Sicherheitsstruktur nutzen wollen, so sollten Sie in diesem Modul die einzelnen Rechte für die verschiedenen Gruppen an Benutzern vergeben.

5.3.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=user	Aufruf über den Administrationsbereich
index.php?file=user	Aufruf über den öffentlichen Bereich
user.html	Aufruf über den öffentlichen Bereich (Suchmaschinenfreundliche URL's)

5.3.2. Avatare

Benutzer können sich selbst in einer 190x145 Pixel großen Avatar-Grafik darstellen. Diese wird im öffentlichen Profil eines Benutzers angezeigt und kann von Besuchern etc. eingesehen werden. Die Avatar-Grafiken werden auf den Webserver geladen und in einer automatisch in der Größe angepassten Version gespeichert. Die Avatar-Funktion kann unter "Benutzer> Mitglieder" deaktiviert werden.

5.3.3. Benutzer löschen

Eingeloggte Benutzer können auf Wunsch ihren jeweiligen Account komplett aus der Datenbank löschen. Dies wird aufgrund der Datenschutzbestimmungen in einigen Ländern der EU notwendig. Eine Abmeldung aus dem Newsletter muss getrennt davon vorgenommen werden. Die Löschung des Accounts ist unter "Benutzerprofil bearbeiten" am Ende der Seite unter "index.php?file=user" möglich.

5.3.4. Detaillierte Rechteverwaltung

Für einzelne Module wie das Artikel-Modul existiert neben der allgemeinen Rechteverwaltung noch eine detaillierte Rechteverwaltung, in der die Verteilung der Rechte auf die einzelnen Gruppen noch feiner granuliert festgelegt werden kann.

Im Artikel-Modul bspw. besteht die Möglichkeit, die Rechte für jede Kategorie einzeln sowie für alle Kategorien insgesamt festzulegen. Für jede Gruppe kann bestimmt werden, welche der folgenden Rechte wo vorhanden sind:

- Artikel hinzufügen
- Artikel bearbeiten
- Artikel freigeben
- Artikel löschen
- Notizen zu einem Artikel schreiben
- Verwendung des WYSIWYG-Editors

Festgelegt wird dies über einen einfachen grafischen Block (siehe Screenshot unten), in welchem ein grünes Feld besagt, dass eine Gruppe über dieses Recht verfügt. Ein rotes Feld hingegen sperrt diese Funktionen für die Gruppe innerhalb der Kategorie. Ein Klick auf eines der Felder ändert den Status. Mit den Buttons auf der rechten Seite einer jeden Zeile können schnell alle Rechte für eine Gruppe gesetzt bzw. blockiert werden.



Besondere Beachtung kommt dem Block unter "admin.php?file=article&mode=category" zu, da die hier festgelegten Rechte allgemein für alle Kategorien gültig sind. Über die detaillierte Rechteverwaltung ist es in Kombination mit der allgemeinen Rechteverwaltung und der Zuteilung der Benutzer auf einzelne Gruppen somit einfach möglich, den Zugriff gezielt auf bestimmte Informationen zu begrenzen (z.B. für Sportreporter nur mit Zugriff auf die Kategorie "Sport" und

ohne Rechte zur Freigabe).

5.3.5. Login / Logout

Um als angemeldeter Benutzer in den Administrations-Bereich zu wechseln, muss der Login über "index.php?file=user" durchgeführt werden bzw. über den Container "{cp:container module=user}". Als Angaben sind der Benutzername und dass Passwort notwendig, wie vorab bei der Registrierung festgelegt. Bei aktiviertem Sicherheitscode muss zusätzlich der Text aus der generierten Grafik in ein Eingabefeld übernommen werden.

5.3.6. Mitgliederliste

Im öffentlichen Bereich kann unter "index.php?file=user&mode=memberlist" eine Mitgliederliste zusammen mit einer Suchfunktion abgerufen werden. Es werden hier nur Mitglieder aufgelistet, welche dies auch in ihrem Profil aktiviert haben. Es werden zu jedem Mitglied allgemeine Informationen wie die Email-Adresse sowie ein Link zu einer eventuell vorhandenen Homepage angezeigt. Weiterhin sind auf der Liste Buttons für verschiedene Messenger-Programme wie ICQ, AOL Instant Messenger oder Yahoo Messenger sichtbar, sofern diese im Profil angegeben werden konnten. Von der Mitgliederliste kann man über den Namen der Mitglieder zu einem erweiterten Profil gelangen.

5.3.7. Mitgliederoptionen

Unter "Benutzerverwaltung" können gezielt Eingabefelder für die Benutzer freigeschalten bzw. gesperrt werden. Diese Einstellungen werden automatisch auf das öffentliche Profil, die Anmeldung sowie die Bearbeitung des Profiles durch den jeweiligen Benutzer übertragen.



Je nach Ausrichtung der Webseite kann es sinnvoll sein, das Mitgliedersystem auf wenige Eingabefelder zu beschränken. Vor allem bei kleineren Mitgliederkreisen, welche direkt an der Gestaltung der Webseite beteiligt sind, werden nicht die vollen Funktionen benötigt. Communities setzen im Gegensatz dazu auf den Austausch zwischen den einzelnen Mitgliedern. Hierfür lassen sich zahlreiche Kontaktmöglichkeiten in Form von Email-Adresse, Messenger-Nummern usw. freischalten.

Unabhängig von den ausgewählten Eingabefeldern sind immer die Eingabefelder für Benutzername und Passwort vorhanden. Weiterhin gibt es Wahlmöglichkeiten für Sprache und Vorlage der Webseite (sofern weitere Optionen vorhanden sind) sowie für die öffentliche Anzeige von Status und Email-Adresse des Benutzers.

5.4. CP::Community

*CP::Community ist ein **kostenpflichtiges Zusatzmodul** für **Contentpapst**. Diese Informationen benötigen Sie daher nur, wenn Sie CP::Community lizensiert haben und auf Ihrer Webseite einsetzen wollen. Weitere Informationen zu CP::Community erhalten Sie unter http://www.sandoba.de/produkte/forum-cpcommunity/.*

5.4.1. Installation von CP::Community

CP::Community wird zusätzlich zu einer bestehenden Contentpapst-Lizenz installiert. Das gelieferte ZIP-Archiv wird dafür zuerst extrahiert (inkl. der dort enthaltenen Ordner). Anschließend wird folgendermaßen vorgegangen:

- 1. Den Ordner "forum" aus dem Archiv per FTP in den Ordner "modules" hochladen (d.h. als Unterordner "/modules/forum/").
- 2. Als Administrator in Contentpapst einloggen und unter "Einstellungen > Modulaktivierung" wechseln.
- 3. Beim Eintrag für CP::Community auf "Installieren" klicken.

Mit dieser Aktion wird dann die Datenbank von Contentpapst entsprechend angepasst, das Modul wird aktiviert und Sie können nun direkt im Anschluß über "Module > CP::Community" bzw. direkt über die URL "admin.php?file=forum" zu diesem Modul wechseln. Das Forum kann im öffentlichen Bereich über die URL "index.php?file=forum" aufgerufen werden.

Wie Sie Ihr Forum den eigenen Wünschen gemäß anpassen und z.B. auch die Suchmaschinenfreundlichen URL's für das Forum aktivieren können, erfahren Sie in den weiteren Einträgen in diesem Abschnitt der Anleitung.

5.4.2. Anpassung der Vorlagen für die Foren, Kategorien, Beiträge usw.

Für die Darstellung der einzelnen Elemente des Forums werden zahlreiche Vorlagen genutzt, welche eine starke Differenzierung der Gestaltung ohne weitere Programmierung ermöglichen. Die Vorlagen sind standardmäßig im Ordner "/modules/forum/templates/" zu finden. Folgende Vorlagen werden mitgeliefert:

- **"addpost_header.html"**
  Darstellung der Beiträge bei der Beantwortung eines Themas (Kopfbereich).
- **"addpost_simple.html"**
  Darstellung der Beiträge bei der Beantwortung eines Themas.
- **"category.html"**
  Darstellung einer Kategorie, welche bereits Beiträge enthält.
- **"category_empty.html"**
  Darstellung einer Kategorie, welche noch keine Beiträge enthält.
- **"container_list_topics.html"**
  Darstellung der Themen für den Container "list_topics".
- **"container_statistic.html"**
  Darstellung der Foren-Statistik für den Statistik-Container.

- **"maintenance.html"**
  Vorlage, welche angezeigt wird, wenn der Wartungsmodus aktiviert ist.
- **"format_actionscript.html"**
  Darstellung von in Beiträgen eingebettetem Actionscript-Code.
- **"format_edited.html"**
  Darstellung des Hinweises zu einem bearbeiteten Beitrag.
- **"format_php.html"**
  Darstellung von in Beiträgen eingebettetem PHP-Code.
- **"format_quotation.html"**
  Darstellung von in Beiträgen eingebetteten Zitaten.
- **"format_sql.html"**
  Darstellung von in Beiträgen eingebettetem SQL-Code.
- **"list_category.html"**
  Darstellung einer Kategorie in der Foren-Übersicht.
- **"list_post.html"**
  Darstellung einer Beitrages in der Themen-Ansicht.
- **"list_post_search.html"**
  Darstellung einer Beitrages in den Suchergebnissen.
- **"list_section.html"**
  Darstellung eines Forums in der Foren-Übersicht.
- **"list_topic.html"**
  Darstellung einer Themas in der Kategorien-Ansicht.
- **"list_topic_announcement.html"**
  Darstellung einer Themas in der Kategorien-Ansicht(Ankündigung).
- **"misc.html"**
  Allgemeine Darstellung "sonstiger" Seiten wie z.B. der Statistik.
- **"search.html"**
  Darstellung des Suchformulares samt der Ergebnisseiten.
- **"site.html"**
  Allgemeine Darstellung wichtiger Foren-Seiten wie z.B. der Startseite.
- **"topic.html"**
  Darstellung eines Themas samt der einzelnen Beiträge.
- **"style.css"**
  Für die Farben und Formatierungen des Forums verantwortliche CSS-Datei.

An der Vorlage "main.htm" aus dem Vorlagenset ist zusätzlich noch eine Anpassung vorzunehmen, damit die CSS-Befehle zur Gestaltung des Forums verarbeitet werden können. Folgende Zeile ist dort in den <head>-Bereich einzubinden:

```
<link rel="stylesheet" href="/{cp:configuration:main:directory}modules/forum/templates/style.css" type="text/css"
media="screen, print" />
```

5.4.3. Container: Ausgabe von statistischen Angaben zum Forum

Über den Container "statistic" des Foren-Moduls können über eine Vorlage an einer beliebigen Stelle der Webseite statistische Angaben zu den Foren-Aktivitäten ausgegeben werden. Dazu muss zunächst an der gewünschten Stelle (z.B. in einem Artikel oder in einer Artikelvorlage) der folgende Containeraufruf platziert werden:

```
{cp:container module="forum" type="statistic"}
```

Weitere Parameter können bei diesem Container nicht übermittelt werden. Der Container lädt dann die Vorlage "modules/forum/templates/container_statistic.html". In dieser können die folgenden {cp:tags} platziert werden, um die einzelnen statischen Angaben anzuzeigen:

{cp:tags} für die Vorlagen des Containers	
{cp:forum:statistic:num_categories}	Anzahl der derzeit vorhandenen Kategorien.
{cp:forum:statistic:num_days}	Anzahl der Tage, seit dem Start des Forums (bzw. der Contentpapst-Statistik).
{cp:forum:statistic:num_posts}	Anzahl der derzeit vorhandenen Beiträge.
{cp:forum:statistic:num_postsperday}	Anzahl der Beiträge pro Tag seit dem Start.
{cp:forum:statistic:num_topics}	Anzahl der derzeit vorhandenen Themen.
{cp:forum:statistic:num_user}	Anzahl der derzeit vorhandenen Benutzer.

5.4.4. Container: Dropdown-Navigationsmenü der Foren / Kategorien anzeigen

Dieser Container wird standardmäßig in den Vorlagen des Forums eingesetzt. Der Container "list_categories" erzeugt ein Dropdown-Navigationsmenü, welches innerhalb des Forums einen schnellen Wechsel zu anderen Kategorien und Beiträgen ermöglicht. Allerdings kann der Container genauso in allen anderen Bereichen von Contentpapst eingesetzt werden (z.B. direkt auf der Startseite einer Webseite):

```
{cp:container module="form" type="list_categories"}
```

Weitere Parameter können bei diesem Container nicht übermittelt werden. Die Anpassung der Darstellung der Foren und Kategorien innerhalb des Dropdown-Navigationsmenüs ist über die CSS-Klassen "forum_dropdown_sections" und "forum_dropdown_categories" in der für das Forum zuständigen CSS-Datei "/modules/forum/templates/style.css" möglich.

5.4.5. Container: Themen einer Kategorie bzw. allgemein anzeigen

Gerade wenn Sie CP::Community im Rahmen einer größeren Webseite einsetzen, kann es oftmals vorteilhaft sein, wenn auch andere Bereiche der Webseite mit dem Forum als zentraler Kommunikationsplattform verknüpft werden. Besucher anderer Bereiche Ihrer Webseite können so direkt erfahren, welche Diskussionen es im Forum zum Thema dieses

Bereiches gibt etc. Genau diese Verknüpfung ermöglicht der Container "list_topics" von CP::Community.

Dazu muss zunächst mindestens der folgende Containeraufruf in einer beliebigen Vorlage oder aber z.B. einem Artikel platziert werden:

```
{cp:container module="forum" type="list_topics"}
```

Über den Parameter "template" kann eine alternativ zu verwendende Vorlage festgelegt werden, standardmäßig wird auf die Vorlage "/modules/forum/templates/container_list_topics.html" zugegriffen. Mit dieser Standardvorlage wird dann eine Auflistung der neuesten Themen im Forum ausgeben.

Folgende Parameter sind für diesen Container zulässig / erforderlich:

Parameter für diesen Container		
category	ID einer Kategorie, um nur Themen aus dieser Kategorie anzuzeigen	optional
dir	Ordnungsrichtung ("asc"= aufsteigend, "desc" = absteigend)	optional
limit	Anzahl der insgesamt angezeigten Einträge (standardmäßig 10 Einträge)	optional
template	Vorlage, welche für die Anzeige verwendet wird	optional
order	Feld nach dem geordnet werden soll ("number", "topic", "category", "title", "date", "lastedit_date")	optional

5.4.6. Darstellung der Beiträge durch Beitragssymbole / Icons und Emoticons auflockern

Beim Anlegen von Themen und Beiträgen können die Benutzer verschiedene Grafiken wählen, mit denen sie anderen mehr über den Beitrag und dessen Inhalt an sich (per Icon) und ihre Launen (per Emoticon) mitteilen können. Die gewünschten Grafiken können dabei einfach per Klick innerhalb dieser beiden Formulare ausgewählt werden. Die Beitragssymbole werden standardmäßig u.a. in der Auflistung der Beiträge innerhalb einer Kategorie angezeigt. Die Icons erscheinen hingegen direkt innerhalb der Beiträge.

Damit beim Anlegen von Themen und Beiträgen derartige Grafiken ausgewählt werden können, müssen Sie dazu die dafür vorgesehenen Grafiken in den Ordnern "/modules/forum/images/icons/" (Icons) und "/modules/forum/images/smilies/" (Emoticons) ablegen. Alle Dateien mit den Endungen "JPEG", "JPG", "GIF" und "PNG" werden automatisch in den Formularen mit angezeigt. Obwohl auch JPEG-Dateien unterstützt werden, sind in diesem Fall aufgrund der geringeren Dateigrößen und der höheren Kantenschärfe jedoch allgemein "GIF"- und "PNG"-Dateien die bessere Wahl.

5.4.7. Suchmaschinenfreundliche URL's aktivieren

CP::Community ist bereits standardmäßig für optimale Suchmaschinenfreundlichkeit vorbereitet. Dazu gehört u.a. auch die Simulation von statischen Seiten mit der Endung ".html". Dies wird über eine Erweiterung des Apache-Webservers mit dem Namen "mod_rewrite" möglich. Ausführliche Informationen finden Sie u.a. in "Systemkomponenten / Module > Einstellungen > Suchmaschinenfreundliche URL's" und "FAQ: Kann ich meine Seite gezielt für Suchmaschinen optimieren?".

Zusätzlich zu den dort genannten Schritten zur Aktivierung der suchmaschinenfreundlichen URL's benötigt CP::Community einige weitere Regeln, die nach der Aktivierung in der Datei ".htaccess" zu platzieren sind (unterhalb der standardmäßigen RewriteRules):

```
RewriteRule ^forum-(add|delete)-topic-([0-9]+)\.html \index.php?file=forum&mode=$1_topic&form[number]=$2 [L]

RewriteRule ^forum-(add|edit|delete)-post-([0-9]+)\.html \index.php?file=forum&mode=$1_post&form[number]=$2
[L]

RewriteRule ^forum-c([0-9]+)-([a-zA-Z0-9-]+)-([0-9]+)\.html \index.php?file=forum&mode=showcategory&form
[category]=$1&form[page]=$3&mod_rewrite=1 [L]

RewriteRule ^forum-c([0-9]+)-([a-zA-Z0-9-]+)\.html \index.php?file=forum&mode=showcategory&form[category]
=$1&mod_rewrite=1 [L]

RewriteRule ^forum-t([0-9]+)-([a-zA-Z0-9-]+)-([0-9]+)\.html \index.php?file=forum&mode=showtopic&form[topic]
=$1&form[page]=$3&mod_rewrite=1 [L]

RewriteRule ^forum-t([0-9]+)-([a-zA-Z0-9-]+)\.html \index.php?file=forum&mode=showtopic&form[topic]
=$1&mod_rewrite=1 [L]

RewriteRule ^forum-s([0-9]+)-([a-zA-Z0-9-]+)-([0-9]+)\.html \index.php?file=forum&form[section]=$1&form[page]
=$3&mod_rewrite=1 [L]

RewriteRule ^forum-s([0-9]+)-([a-zA-Z0-9-]+)\.html \index.php?file=forum&form[section]=$1&mod_rewrite=1 [L]

RewriteRule ^forum-([a-z]+)\.html \index.php?file=forum&mode=$1&mod_rewrite=1 [L]
```

Die URL's innerhalb der Vorlagen werden dann vom System automatisch im neuen Format generiert, sobald die Suchmaschinenfreundlichen URL's in den Einstellungen aktiviert wurden.

5.5. CP::Forms

Individuelle Datenbanken und Verzeichnisse (z.B. Download-Archive, Jobverzeichnisse) können einfach mit CP::Forms erstellt werden. Folgen Sie dem Wizard und Sie erhalten ein fertiges Modul mit allen Konfigurations-Dateien für die Formulare.

5.5.1. Allgemeines zu diesem Modul

Mit CP::Forms steht in Contentpapst ein sehr leistungsstarkes Modul zur Erstellung von Formularen und neuen Modulen zur Verfügung. Alle Aktionen zum Hinzufügen / Bearbeiten dieser Module werden dabei direkt über die grafische Oberfläche des Systems durchgeführt. Die mit CP::Forms erstellbaren Module sind äußerst vielfältig. Dies können z.B. folgende sein:

- Markt- und Meinungsforschung
- Support-, Kontakt- und Bugtracking-Formulare
- Jobdatenbanken, Pressematerialien
- Immobilien, Link-Archive, Branchenbücher
- uvm.

Die einzelnen Module werden dabei in CP::Forms grundsätzlich in sogenannten "Collections" gebündelt. Diese "Collections" sind thematische Sammlungen von Modulen (bspw. für einen Veranstaltungskalender Module für die Kategorien und die Einträge) und ermöglichen eine bessere Übersicht.

Über "Modul hinzufügen" bzw. "Modul aus Vorlage erstellen" wird in CP::Forms ein neues Modul angelegt. Bei der letzteren Möglichkeit wird ein bereits bestehendes CP::Forms-Modul als Grundlage für die Erstellung eines neuen Modules verwendet.

Im Normalfall wird jedoch ein komplett neues Modul erstellt. Für dieses Modul müssen zuerst grundlegende Angaben gemacht werden, die Titel und Beschreibung sowie den internen Modul-Namen umfassen. Weiterhin muss festgelegt werden, um welche Art von Modul es sich handelt (ein Modul zur Verwaltung von Einträgen oder zum Versand von Emails [z.B. in Form eines Kontaktformulares]). Entsprechend dieser Festlegung werden weitere Angaben erforderlich.

Im nächsten Schritt der Modulerstellung werden die für das Modul benötigten Felder zusammengestellt. Schrittweise werden die Feldtypen gewählt und die jeweils angezeigten Eingabefelder gefüllt, u.a. mit Angaben zum internen Feldnamen und zur Bezeichnung, der Feldlänge, Sichtbarkeit usw.

Sobald dieser Schritt abgeschlossen wurde, legt CP::Forms im Hintergrund alle Dateien des neuen Modules in einem neuen Ordner unter "/modules/[ORDNER-NAME]/" an und es kann sofort mit dem Einpflegen der Inhalte begonnen werden. Die Darstellung der Inhalte eines solchen Modules erfolgt dabei stets über die Container. Dabei werden die standardmäßig verfügbaren Container in weiteren genauer beschrieben.

Vorab noch ein Hinweis: Wenn auch im öffentlichen Bereich Formulare zum Hinzufügen neuer Einträge angezeigt werden sollen, dann muss im Quelltext der Seite die CSS-Datei für CP::Forms eingebunden werden. Diese sorgt für eine ordnungsgemäße Anzeige der einzelnen Formularfelder. Kopieren Sie dazu die Datei "cpforms.css" aus dem Ordner "/themes/backend/" in Ihren eigenen Vorlagenordner und binden Sie die CSS-Datei über folgende Zeile im Kopfbereich der Vorlage "main.htm" ein:

```
<link rel="stylesheet" href="/{cp:configuration:main:directory}{cp:path}cpforms.css" type="text/css" media="screen,
print" />
```

Wichtige URL's für dieses Modul	
admin.php?file=forms	Aufruf über den Administrationsbereich

5.5.2. Besonderheiten bei der Feld- und Modulbenennung

Wenn Sie ein neues Modul über CP::Forms erstellen, müssen Sie bei der Bezeichnung der Datenfelder auf Ihre Eingaben achten. Erlaubt sind hier nur alphanumerische Zeichen ("A-Z", "a-z", "0-9"), wobei am Anfang eines Feldnamens keine Zahl stehen darf. Weiterhin sollten nur Kleinbuchstaben verwendet werden. Contentpapst nimmt selbstständig im Hintergrund Konvertierungen vor, jedoch sollten Sie auch selbst auf diese Besonderheit achten, damit das Modul korrekt erstellt werden kann. Die gleiche Einschränkung besteht bei der Wahl des Ordner-Namens für ein Modul.

5.5.3. Container: Formular zum Eintragen von Inhalten

Natürlich dient CP::Forms nicht nur der bloßen Eingabe von Inhalten im Administrationsbereich und der Ausgabe dieser Inhalte im öffentlichen Bereich. Vielmehr lassen sich auch Interaktionen zwischen beiden Bereichen herstellen, u.a. durch ein Formular über welches Besucher neue Einträge hinzufügen oder (wenn Einträge nicht in der Datenbank gespeichert werden) Emails versenden können.

Ermöglicht wird diese Funktionalität über einen Container mit dem Typ "form". Dieser generiert automatisch das Formular samt aller notwendigen client- und serverseitigen Überprüfungen.:

```
{cp:container module="[MODULNAME]" type="form" [return="status.php"]}
```

Handelt es sich um ein Modul zur Verwaltung von Einträgen in der Datenbank, so wird mit dem Absenden des Formulares automatisch ein neuer Eintrag in der Datenbank erzeugt. Dieser muss anschließend durch berechtigte Benutzer im Administrationsbereich freigegeben werden. Handelt es sich hingegen um ein Formular zum Versand von Emails (z.B. ein Kontaktformular), so wird der Inhalt des Formulares per Email versandt.

Neben den erforderlichen Parametern "module" und "type" gibt es für den "form"-Container den optionalen Parameter "return". Über diesen kann eine URL angegeben werden, auf die nach dem Abschicken gewechselt werden soll. Alternativ kann der Parameter "return_article" mit der ID eines Artikels genutzt werden. Ohne Angabe dieser Parameter wird auf die ursprüngliche Seite zurückgekehrt. Beachten Sie dabei, dass in der URL das Zeichen "=" stets durch "=" zu ersetzen ist, damit der Container korrekt verarbeitet werden kann.

Wenn ein derartiges Formular mit CP::Forms-Inhalten oder anderen Inhalten aus Contentpapst verknüpft werden soll, dann kann dafür der modulspezifische Parameter "field_[FELDNAME]" beim Containeraufruf genutzt werden. Ein Beispiel für den Einsatz dieses Features ist die Übergabe eines Produktnamens im Containeraufruf an ein per CP::Forms generiertes Formular, über welches der Kunde Kontakt mit dem Hersteller aufnehmen kann. Damit der Kunde den Produktnamen des

gerade aufgerufenen Produktes nicht mehr manuell eingeben muss, kann dieser direkt beim Containeraufruf mit übergeben werden:

```
{cp:container module="forms_kontaktformular" type="form" field_title="{cp:forms:values:title}"}
```

Der obenstehende Containeraufruf könnte z.B. in der Detailansicht eines Produktes integriert werden, dessen Daten über CP::Forms gepflegt werden. Wenn dort ein Feld mit dem Namen "title" vorhanden ist, dann wird über "{cp:forms:values:title}" der Titel des angezeigten Produktes an den Container übergeben. D.h. dem Container "form" des CP::Forms-Modules "forms_kontaktformular" wird dieser Produktname direkt mit übermittelt, so dass dieser im Formular weiterverarbeitet werden kann. Statt des Tags "{cp:forms:values:title}" kann aber natürlich auch jeder andere {cp:tag} bzw. auch ein ganz normaler Text übergeben werden.

5.5.4. Container: Auflistungen auf der Webseite anzeigen

Eine Auflistung von einzelnen Einträgen eines mit CP::Forms erstellten Modules wird wie gewohnt über einen Container sowie die {cp:tags} erstellt. Dabei kann über die Parameter des Containers frei angegeben werden, wie die angezeigten Einträge formatiert (Parameter "template") und geordnet sein sollen (Parameter "order" und "dir" ["asc"/"desc"] für die Richtung), wieviele Einträge max. ausgegeben werden sollen (Parameter "limit") und welche Seite angezeigt wird (Parameter "page").

Dieser Container gibt z.B. die ersten 5 Einträge des Modules "jobs" aus:

```
{cp:container module="jobs" type="list" page="1" limit="5" order="date_added" dir="asc"
template="modules/jobs/templates/job_vorlage.html"}
```

Die Einträge sind im Beispiel nach dem Datum der Eintragung geordnet (aufsteigend). Zur Anzeige der Einträge wird die Vorlage "job_vorlage.html" im Unterordner "templates" des Modules verwendet.

In der über den Parameter "template" angegebenen Vorlage können jeweils die in der "{cp:tags}-Erklärung" des Modules genannten {cp:tags} verwendet werden (diese "{cp:tags}-Erklärung" kann direkt aus der Übersichtsseite des Modules heraus aufgerufen werden). Wenn keine Vorlage angegeben wird, so wird die Vorlage "list_public.html" im Unterordner "templates" des Modules standardmäßig genutzt.

Über die Parameter "select_field" und "select_value" ist optional eine Filterung der Einträge möglich. Für den Parameter "select_field" wird der Feldname angegeben, für "search_value" der gewünschte Wert. Wenn in mehreren Feldern gesucht werden soll, dann sind die Feldnamen / Werte durch Kommata zu trennen. Eine Ähnlichkeitssuche ist über "%" bei einem Wert oder mehreren Werten möglich:

```
{cp:container module="events" type="list" select_field="section" select_value="2"}
// Einschränkung der Auflistung auf die Kategorie mit der ID "2"
```

```
{cp:container module="events" type="list" select_field="location_city" select_value="Berl%"}
// Einschränkung auf Veranstaltungsorte mit dem Stamm "Berl" (also z.B. "Berlin", "Berlingen"...)
```

Folgende Angaben sind zulässig / erforderlich (unterstrichen):

Parameter für diesen Container	
date_from	Einträge ab ... anzeigen (Zeitangabe im Timestamp-Format erforderlich)
date_to	Einträge bis ... anzeigen (Zeitangabe im Timestamp-Format erforderlich)
dir	Aufsteigend ("asc") oder absteigend ("desc") ordnen
limit	Anzahl der insgesamt angezeigten Einträge
max_age	Max. Alter der anzuzeigenden Einträge in Sekunden
order	Einträge ordnen (siehe unten)
page	Angezeigte Seite
select_field	Feldname für die Filterung der Einträge
select_value	Wert für die Filterung der Einträge
template	Pfad zu einer Vorlage (andernfalls wird Standardvorlage genutzt)
type	Typ des Containers (hier immer "list")

Werte für den Parameter "order" (Ordnung)	
number	Nach Eintragungsreihenfolge ordnen
hits	Nach Anzahl der Aufrufe ordnen
version	Nach Versionsnummer ordnen
date_added	Nach Datum der Eintragung ordnen
date_lastmodified	Nach Datum der letzten Änderung ordnen
random	Zufällig ordnen
start_time	Laufzeit Start
stop_time	Laufzeit Stop
status	Status
...	Zusätzlich kann für "order" jeder Feldname des Modules angegeben werden.

5.5.5. Container: Navigation zum Umblättern erzeugen

Der Container zur Auflistung von Einträgen eines Modules bietet u.a. den Parameter "page" für die Auswahl einer

bestimmten Seite an. Damit kann die Auflistung der Einträge auf mehrere Seiten aufgeteilt werden, indem die Nummer der gewünschten Seite als Parameter in der URL übergeben wird (z.B. "index.php?id=__ID__&showpage=4").

Damit dann auch die Einträge dieser Seite angezeigt werden, muss der Parameter "page" im Aufruf des Containers mit "... page="{cp:variable:showpage}" ..." weitergegeben werden.

Doch wie kann nun zwischen den verfügbaren Seiten umgeschalten werden, ohne dass per Hand oder per fester Verlinkung die unterschiedlichen URL's aufgerufen werden müssen? Für dieses Problem gibt es den Container "pages". Mittels dieses Containers kann für ein Modul automatisch eine Navigation generiert werden (à la "1 2 3 4" usw.):

```
{cp:container module="__MODULNAME__" type="pages" page="{cp:variable:showpage}" varname="showpage"
per_page="10"}
```

Zwingend anzugeben sind für diesen Container die Parameter "module" (der Name des Modules, für den die Navigation erstellt wird), "type" (immer "pages") sowie die Parameter "page", "varname" und "per_page".

Als Parameter "page" wird - wie beim ersten Container auch - der Wert aus der URL übergeben. Da der Name dieses Parameters frei gewählt werden kann, wird dem Container über den Parameter "varname" mitgeteilt, welcher Parameter in den URL's zum Aufbau der Navigation zu verwenden ist. Der Parameter "per_page" schließlich gibt an, wieviele Einträge pro Seite angezeigt werden. Dieser muss daher den gleichen Wert aufweisen wie der Parameter "limit" beim ersten Container. Über den Parameter "max_pages" können Sie festlegen, wieviele Seiten maximal vor/nach der aktuellen Seite in der Navigation sichtbar sein sollen (standardmäßig sind dies 15 Seiten).

Eine Auflistung von Einträgen eines Modules samt Navigation könnte also z.B. den folgenden Aufbau aufweisen:

```
{cp:container module="veranstaltungen" type="list" order="date_added" dir="desc" limit="5"
page="{cp:variable:showpage}"}
{cp:container module="veranstaltungen" type="pages" page="{cp:variable:showpage}" varname="showpage"
per_page="5"}
```

5.5.6. Container: Einzelne Einträge auf der Webseite anzeigen

Neben der Anzeige des Formulares zum Hinzufügen von Einträgen und der Auflistung der Einträge auf der Webseite gibt es auch die Möglichkeit, gezielt einen einzelnen Eintrag anzuzeigen. Jedes mit CP::Forms erstellte Modul hat eine Spalte mit der Kennzeichnung "ID". Dies ist die Nummer, anhand derer ein Eintrag ausgewählt werden kann. Für die Anzeige auf der Webseite müssen Sie also den Namen des mit CP::Forms erstellten Modules und die Nummer des gewünschten Eintrages kennen.

An der Stelle, an welcher später der Eintrag sichtbar sein soll, schreiben Sie dann z.B. folgendes:

```
{cp:container module="jobs" type="details" number="2" template="job_vorlage.htm"}
// Eintrag mit der ID"2" aus dem Modul "jobs"
```

Im obenstehenden {cp:tag} wurde zusätzlich eine Vorlage angegeben. In dieser Vorlage wird die Formatierung des Eintrages festgelegt. Alle Felder des Modules / der Datenbank können angezeigt werden. Der Tag für jedes Feld ist immer im Format "{cp:forms:values:__FELDNAME__}" aufgebaut. Genauere Informationen über die jeweils verfügbaren {cp:tags} können in der "{cp:tags}-Erklärung" des Modules abgerufen werden.

Der Parameter "template" ist im Aufruf des Containers optional. Wenn dieser Parameter weggelassen wird, dann wird standardmäßig die Vorlage "details_public.html" aus dem Unterordner "templates" eines Modules genutzt.

5.5.7. Container: Suchfunktion im öffentlichen Bereich

Die zentrale Ansicht eines Modules im Administrationsbereich mit Suchfunktion und der Möglichkeit zum Blättern zwischen den Seiten lässt sich als Container auch im öffentlichen Bereich darstellen. Dazu muss im einfachsten Fall folgender Containeraufruf genutzt werden:

```
{cp:container module="[MODULNAME]" type="search" template="[PFAD ZUR VORLAGE]"}
```

Angezeigt wird dann über die Vorlage "search.html" (im Vorlagen-Ordner des Modules) ein Suchformular, über welches die Benutzer innerhalb der einzelnen Felder suchen können. Die Einträge werden dabei mittels der über den Parameter "template" angegebenen Vorlage dargestellt. Die in der Vorlage "search.html" einsetzbaren {cp:tags} können direkt in der "{cp:tags}-Erklärung" bei der Modulansicht eingesehen werden.

```
{cp:container module="[MODULNAME]" type="search" [limit="10"] [order="number"] [dir="asc"]
[template="vorlage.html"] [format="html"]}
```

Oben sehen Sie den {cp:tag} noch einmal in einer erweiterten Fassung. Hier werden Standardwerte für die Anzahl der Einträge pro Seite (Parameter "limit"), die Ordnung ("order"), die Ordnungsrichtung ("dir"), die Vorlage ("template") und das Format ("format") vorgegeben.

5.5.8. Container: Erweiterte Suchfunktion im öffentlichen Bereich

Wenn Sie für ein datenbank-basiertes CP::Forms-Modul eine Suchfunktion im öffentlichen Bereich anbieten möchten, dann steht Ihnen neben der einfachen Volltextsuche (wie in "Suchfunktion im öffentlichen Bereich" näher erläutert) auch eine erweiterte Suchfunktion zur Verfügung. Wichtig: Damit die Einträge über diese Suchfunktion durchsucht werden können, müssen die Einträge zunächst über "Modul bearbeiten" in eine eigene Datenbanktabelle ausgelagert werden.

Diese Suchfunktion mit dem Namen "prosearch" gibt dem Besucher z.B. bei einem Veranstaltungsmodul die Möglichkeit, gezielt nach Veranstaltungen mit einem bestimmten Namen, aus einer Kategorie, zu einem speziellen Thema usw. zu

suchen. Das dafür erforderliche Formular ist auch im Administrationsbereich über den Link "Erweiterte Suche" erreichbar und wird in beiden Bereichen automatisch von CP::Forms erzeugt.

Um die erweiterte Suchfunktion für ein solches Modul auf der Webseite anzuzeigen, müssen Sie den folgenden Containeraufruf in der gewünschten Vorlage oder beispielsweise einem Artikel einsetzen:

```
{cp:container module="[MODULNAME]" type="prosearch" template="[PFAD ZUR VORLAGE]"}
```

Damit die Suchfunktion auf der Webseite ausgegeben wird, muss zwingend eine Vorlage mit dem Namen "prosearch.html" im Ordner "templates" des CP::Forms-Modules vorhanden sein. Alternativ können Sie über den Parameter "template" auch einen anderen Pfad zu dieser Vorlage angeben. In dieser Vorlage können u.a. diese speziellen {cp:tags} genutzt werden:

{cp:tags} für die Vorlagen des Containers	
{cp:forms:search:form}	HTML-Code für das Suchformular
{cp:forms:search:content}	Auflistung der Suchergebnisse
{cp:forms:search:numresults}	Anzahl der Suchergebnisse
{cp:forms:search:pages}	Auflistung der verfügbaren Ergebnisseiten, Navigation zum Umblättern

Unten sehen Sie den {cp:tag} noch einmal in einer erweiterten Fassung. Hier werden Standardwerte für die Anzahl der Einträge pro Seite (Parameter "limit"), die Ordnung ("order"), die Ordnungsrichtung ("dir"), die Vorlage ("template") und das Format ("format") vorgegeben.

```
{cp:container module="[MODULNAME]" type="prosearch" [limit="10"] [order="number"] [dir="asc"]
[template="vorlage.html"] [format="html"]}
```

5.5.9. Neue Einträge als Email verschicken

Sofern ein Modul öffentlich zugänglich ist und keine Datenbank nutzt, haben die Besucher die Möglichkeit, über das öffentliche Formular (den Container mit dem Typ "form") einen Eintrag hinzuzufügen. Dieser wird jedoch nicht in die Datenbank integriert, sondern als Email verschickt an die bei der Modulerstellung angegebene Email-Adresse. Diese Email wird dabei immer im Textformat ohne HTML-Formatierungen versandt.

Besonders geeeignet ist der Versand von Emails bei:

- Kontaktformularen
- Formularen für die Registrierung / Bestellung (z.B. von Theater-Karten oder DVD-Verleih)
- Feedback-Formularen
- Bewerbungsformularen
- ...

Die Vorlage "email.html" innerhalb des Vorlagen-Ordners eines per CP::Forms erstellten Modules beschreibt, wie die Email aufgebaut ist. Es können dort alle Tags für die Feldinhalte und Feldbezeichnungen verwendet werden wie auch in den anderen Vorlagen. Genauere Informationen erhalten Sie über die "{cp:tags}-Erklärung" auf der Hauptseite eines Modules.

5.5.10. Zugriff auf die Inhalte mittels Webservices

CP::Forms stellt für den extenen Zugriff die Inhalte als XML direkt mittels Webservices zur Verfügung. Um die Webservices zu aktivieren, muss jedoch vorher beim Anlegen eines neuen Modules oder beim Bearbeiten eines vorhandenen Modules die entsprechende Option aktiviert werden. Dann ist der Zugriff über die Datei "webservices.php" möglich.

Grundsätzlich werden dabei nur freigeschaltete Einträge angezeigt. Beim Aufruf über die webservices.php ist stets das gewünschte Modul anzugeben, gefolgt von den einzelnen Parametern für die Angabe von Typ, Nummer des Eintrages etc. D.h. ein Aufruf hat folgendes Format:

```
webservices.php?file=[ORDNER-NAME]&type=[TYP]
```

Standardmäßig kann sowohl auf einzelne Einträge, als auch auf komplette Listen von Einträgen zugegriffen werden. Weitere Abfrage-Typen lassen sich für einzelne Module bzw. global per PHP bereitstellen.

Detailansicht eines Eintrages

Sie benötigen für die Abfrage eines einzelnen Eintrages die ID des Eintrages (die Sie z.B. im Administrationsbereich einsehen können) sowie den Ordner-Namen des Modules. Ein Aufruf kann dann folgendermaßen aussehen:

```
webservices.php?file=forms_test&type=details&number=35
// ruft aus dem Modul "forms_test" den Eintrag mit der ID "35" ab
```

Listenansichten von Einträgen

Um komplette Listenansichten wie im Administrationsbereich bzw. über die Container abzurufen, gibt es den Typ "list". Hierbei sind noch die optionalen Parameter "order" (Ordnung nach "number", "hits", "date_added", "date_lastmodified", jeweils "+" vorangestellt für aufsteigend, "-" für absteigend) und "page" (aktuelle Seite) nutzbar. Pro Seite werden max. 10 Einträge angezeigt.

```
webservices.php?file=forms_test&type=list&order=-date_added&page=2
// ruft aus dem Modul "forms_test" die Einträge auf der 2. Seite ab, geordnet wird mit dem neuesten Eintrag zuerst
```

RSS-Feeds zu einem CP::Forms-Modul

CP::Forms kann zu einem Modul auch automatisch einen vollwertigen RSS-Feed generieren. Dazu muss in der URL der Typ "rssfeed" genutzt werden. Über den Parameter "limit" kann die Anzahl der maximal auszugebenden Einträge festgelegt werden. Standardmäßig werden maximal 15 Einträge im RSS-Feed ausgegeben.

Die Kopfdaten dess RSS-Feeds werden automatisch den Systemeinstellungen entnommen. Dies sind u.a. Angaben wie Titel und Beschreibung der Webseite, die genutzte Sprache, die URL zur Webseite oder genauso das Veröffentlichungsdatum.

Für die einzelnen Einträge werden der Titel dem Feld "title" und die Beschreibung dem Feld "description_short" entnommen, der Link zum ausführlichen Eintrag kommt aus dem Feld "link". D.h. wenn Sie zu einem CP::Forms-Modul (z.B. zu Ihren Pressemeldungen oder Job-Angeboten) einen RSS-Feeds veröffentlichen möchten, dann sollten in diesem Modul Felder mit den oben genannten Feldnamen vorhanden sein.

Der Abruf ist dann über die folgende URL möglich:

```
webservices.php?file=forms_test&type=rssfeed&limit=10
// ruft aus dem Modul "forms_test" max. die 10 neuesten Einträge als RSS-Feed ab
```

5.6. CP::Shop

*CP::Shop ist ein **kostenpflichtiges Zusatzmodul** für **Contentpapst**. Diese Informationen benötigen Sie daher nur, wenn Sie den Shop CP::Shop als Modul von **Contentpapst** bzw. als Standalone-Version einsetzen. Weitere Informationen zu CP::Shop erhalten Sie unter http://www.sandoba.de/produkte/shop-software-cpshop/.*

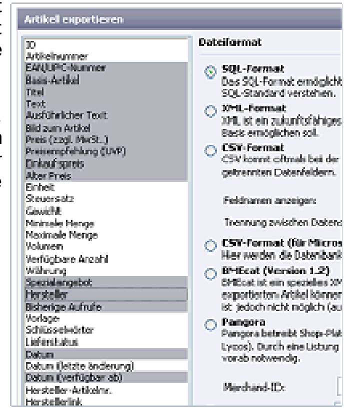
5.6.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=shop	Aufruf über den Administrationsbereich
index.php?file=shop	Aufruf über den öffentlichen Bereich
shop.html	Aufruf über den öffentlichen Bereich (Suchmaschinenfreundliche URL's)

5.6.2. Artikel exportieren / importieren

CP::Shop bietet den Export und den Import von Artikeln in verschiedenen Formaten an. Dazu gehören u.a. auch Exporte für die Einbindung der Artikel in externe Shopping-Portale, sowie für den Austausch kompletter Artikel-Kataloge.

Artikel exportieren



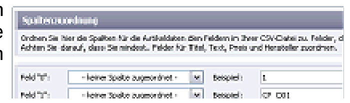
Sie erreichen den Export in CP::Shop unter "Artikel / Artikel exportieren". Dort müssen Sie zunächst auswählen, welche Felder der Artikel Sie in den Export einbinden wollen (dies ist für einige Export-Formate nicht relevant). Über "Alle markieren / Abwählen" können Sie eine Schnellauswahl treffen.

Auf der rechten Seite müssen Sie dann das gewünschte Dateiformat auswählen. Die Standard-Format sind jeweils erklärt. Teilweise sind zusätzliche Angaben notwendig, bspw. für eine Merchant-ID. Weitere Formate stehen Ihnen unter "Alternatives Format" zur Verfügung. Einige mögliche Formate (detaillierte Beschreibungen finden Sie in der Anleitung):

- Shopping-Portal "Billiger.de"
- Shopping-Portal "Evendi"
- Shopping-Portal "Froogle.de"
- Shopping-Portal "Geizhals.at"
- Shopping-Portal "Getprice.de"
- Universal-Datenbank "Google Base" (u.a. für Froogle)
- Shopping-Portal "Hardwareschotte / Schottenland"
- Shopping-Portal "Kelkoo" / "Kelkoo Premium"
- Shopping-Portal "Milando.de"
- Shopping-Portal "Pangora"
- Shopping-Portal "Preisauskunft"
- Shopping-Portal "Preisroboter.de"
- Shopping-Portal "Preissuchmaschine.de"
- Shopping-Portal "Preiswa(h)l"
- Shopping-Portal "Schnaeppchenjagd.de"
- Shopping-Portal "Shoppingscout24.de"
- Shopping-Portal "Shopzilla.de"
- Shopping-Portal "Wir-Lieben-Preise.de"
- Shopping-Portal "Yatego"
- Export als HTML-Liste
- Export im shopinfo.xml-Standard

Anschließend sind in den Optionen noch die Reihenfolge, Ordnungsrichtung, Startnummer und max. Anzahl an Artikeln einstellbar. Diese gelten u.a. für das SQL-, CSV- und XML-Format. Die anderen Formate unterliegen jeweils den Anforderungen der einzelnen Shopping-Portale und können daher nicht im Aufbau beeinflusst werden.

Artikel importieren



Der Artikelimport kann unter "Artikel / Artikel importieren" aufgerufen werden oder alternativ direkt von der Startseite von CP::Shop aus. Als Formate unterstützt der Import momentan die CSV- und XML-Dateien, die z.B. von CP::Shop generiert werden.

Deshalb müssen Sie hier lediglich die Datei und das Format festlegen sowie angeben, ob in der ersten Zeile die Spaltenüberschriften vorhanden sind (dies kann z.B. bei CSV-Dateien der Fall sein).

Anschließend ermöglicht Ihnen CP::Shop anhand eines Beispieldatensatzes die Zuordnung von Feldern aus den Dateien zu den Feldern der Datenbank (siehe Screenshot rechts). Diese Zuordnung sollte sehr sorgfältig durchgeführt werden, da sonst Daten leicht vertauscht werden könnten. Nach dem Abschluss der Zuordnung wird dann der Import der Artikel in das System gestartet.

5.6.3. Artikel exportieren im shopinfo.xml-Standard

Der shopinfo.xml-Standard wurde als Forschungsprojekt innerhalb der Universität Mannheim entwickelt und soll Preisvergleichen und Produktdatenbanken ein möglichst **universell nutzbares Beschreibungsformat für Stamm- und Produktdaten von Online-Shops** bereitstellen. Dieses Format bietet für Shop-Betreiber den Vorteil, dass es mit den erzeugten Dateien eine zunehmende wachsende Anzahl an Portalen versorgen kann.

Für diesen Standard wird eine Datei "shopinfo.xml" benötigt, die allgemeine Informationen über den Shop beinhaltet, sowie eine zweite Datei, die die eigentlichen Artikel beinhaltet. Die Datei "shopinfo.xml" ist dabei generell unter "http://www.ihre-domain.de/shopinfo.xml" abzulegen.

Beide Dateien lassen sich direkt unter "Artikel > Artikel exportieren" als alternatives Format auswählen. Die Artikeldatei müssen Sie dabei nicht generieren lassen. Es genügt, wenn Sie die "shopinfo.xml" über den Export generieren und diese dann am oben genannten Ort ablegen. Alle Abfragen etc. werden dann durch den CP::Shop automatisch beantwortet.

5.6.4. Artikel exportieren zu Billiger.de



Damit Ihre Artikel auch beim Preisvergleichsportal "www.billiger.de" gelistet werden, müssen Sie sich dort einmalig unter http://www.solute.de/partner/ anmelden. Die CSV-basierten Exportdateien können Sie dann entweder unter "Artikel > Artikel exportieren > Alternatives Format: Billiger.de" herunterladen oder aber Sie lassen diese von Billiger.de direkt über die folgende URL von Ihrem Server abrufen:

Artikeldaten im CSV-Format (Billiger.de): */index.php?file=shop&mode=export&clean=yes&type=billiger.de

5.6.5. Artikel exportieren zu Evendi



eVendi informiert über Auswahl, Preise und Bezugsquellen von Tausenden von Produkten aus fast allen Lebensbereichen. Zu jedem Produkt erhält der Besucher eine Übersicht der Preise der registrierten Anbieter und erkennt auf den ersten Blick die günstigsten Preise.

Damit Ihre Artikel bei "www.evendi.de" gelistet werden, müssen Sie sich dort einmalig anmelden. Entscheiden Sie sich bei der Anmeldung für den automatischen Export der Daten und informieren Sie Ihren Ansprechpartner bei Evendi über diese URL:

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=evendi

5.6.6. Artikel exportieren zu Froogle.de



Froogle.de ist ein Angebot des Suchmaschinenbetreibers Google und listet als Preisvergleichsdienst die Artikel zahlreicher Online-Shops. Damit Ihre Artikel bei "www.froogle.de" gelistet werden, müssen Sie sich dort einmalig anmelden. Anschließend können Sie über den Artikel-Export die Artikel-Daten im benötigten Format generieren lassen. Diese Datei muss dann bei Froogle hochgeladen werden, damit diese verarbeitet werden kann:

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=froogle

5.6.7. Artikel exportieren zu Geizhals.at



Der Geizhals-Preisvergleich existiert seit 1996 als unabhängige Preisvergleichs-Plattform im WWW. Auf dieser Seite werden Artikel von Shops aus dem deutschsprachigen Raum gelistet, jeweils auf eigenen Länderseiten. CP::Shop kann für Geizhals.at automatisch alle Artikel im gewünschten Format liefern.

Damit Ihre Artikel bei "http://www.geizhals.at/deutschland/" gelistet werden, müssen Sie sich dort einmalig anmelden. Für Österreich nutzen Sie die Seite "http://www.geizhals.at/" und wenn Sie Ihre Artikel im gesamten EU-Raum anbieten die Seite "http://www.geizhals.at/eu/". Auch ein automatischer Export ist verfügbar unter:

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=geizhals.at

Beachten Sie bitte, dass Geizhals.at besondere Anforderungen an die Artikelbeschreibungen stellt. In diesen soll klar erkenntlich sein, welche auffälligen Merkmale ein Artikel aufweist (Farbe, Größe etc.). Bei Software ist zusätzlich die Sprache anzugeben, bei OEM-Produkten der Hersteller.

5.6.8. Artikel exportieren zu getprice.de

Über den Export zu getprice.de können Sie sich ein weiteres Shopping-Portal für Ihre Produkte erschließen. getprice.de ist vor allem auf größeren Partner-Seiten aktiv. Damit Ihre Artikel bei "www.getprice.de" gelistet werden, müssen Sie sich dort einmalig anmelden. Ein Export im benötigten Format ist unter diesem Link möglich (bzw. über "Artikel > Artikel exportieren

> Alternatives Format: Getprice"):

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=getprice

5.6.9. Artikel exportieren zu Google Base (base.google.de)



Google Base ist die Universal-Datenbank des Suchmaschinenanbieters Google. In Google Base können Inhalte der unterschiedlichsten Art gespeichert werden, von Shop-Produkten über Beschreibungen von Veranstaltungen bis hin zu Rezepten. Mit der Integration der eigenen Artikel in dieser Datenbank werden die Artikel besser auffindbar in den verschiedenen Google-Angeboten (z.B. der Google Produktsuche).

Nachdem die Datei unter "Artikel > Artikel exportieren > Alternatives Format: Google Base" exportiert und lokal gespeichert wurde, muss diese zu Google Base hochgeladen werden. Dafür ist eine vorherige Anmeldung bei Google Base notwendig. Diesen Upload sollten Sie nach Möglichkeit 1x pro Monat wiederholen, da die Angebote nach ca. 30 Tagen von Google gelöscht werden.

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=google_base

5.6.10. Artikel exportieren zu Hardwareschotte / Schottenland

Über den Export zu schottenland.de können Sie sich ein weiteres Shopping-Portal für Ihre Produkte erschließen. Ein Export im benötigten Format ist unter diesem Link möglich (bzw. über "Artikel > Artikel exportieren > Alternatives Format: Schottenland"):

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=schottenland

5.6.11. Artikel exportieren zu Hitmeister



Hitmeister wird betrieben von den Machern von Hitflip - Europas größter Tauschplattform für Mediengüter (Hitflip Media Trading GmbH). Steigern Sie als Verkäufer Ihren Umsatz, indem Sie neue und gebrauchte Artikel auf Hitmeister zum Verkauf anbieten. Das Anbieten der Artikel ist völlig kostenlos, erst wenn Sie Artikel verkaufen und Ihre Zahlung erhalten, fällt eine geringe Provision an. Das Kernelement des Verkaufens auf Hitmeister ist die Sicherheit: Hitmeister bietet Ihnen als Verkäufer 100% Schutz.

Damit Hitmeister alle Angebote in die Datenbank übernehmen kann, müssen die Daten unter "Artikel > Artikel exportieren" im CSV-Format von Hitmeister erzeugt werden. Standardmäßig sind diese Daten zusätzlich unter folgender URL abrufbar:

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=hitmeister

Sie müssen sich jetzt nur noch unter http://www.hitmeister.de/ anmelden und Ihre Shop-Daten eingeben. Für Ihren Shop öffnet sich so eine weitere interessante Plattform.

5.6.12. Artikel exportieren zu kelkoo.de



Kelkoo ist eine effiziente Produktsuchmaschine, die Usern hilft, aus einer großen Auswahl an Online-Shops die besten Angebote zu finden und darüber hinaus Preise zu vergleichen. Als Mittler und neutrale Plattform für Händler und Nutzer schafft Kelkoo damit eine Transparenz und Übersicht des Online-Marktes. Damit Ihre Artikel bei "www.kelkoo.de" gelistet werden, müssen Sie sich dort einmalig anmelden. Je nach gewähltem Paket können Sie die Dateien selbst exportieren und bei Kelkoo hochladen oder diese automatisch austauschen lassen:

Artikeldaten im CSV-Format (Kelkoo): */index.php?file=shop&mode=export&clean=yes&type=kelkoo
Artikeldaten im CSV-Format (Kelkoo Premium): */index.php?file=shop&mode=export&clean=yes&type=kelkoo_premium

5.6.13. Artikel exportieren zu Pangora



Pangora ist ein neutraler Anbieter von Artikeldaten vieler Online-Shops. Diese gesammelten Daten werden u.a. Portalen wie n-tv.de, Lycos, dooyoo und freenet.de zur Verfügung gestellt. Für den Export benötigen Sie zuerst eine Anmeldung bei Pangora (http://www.pangora.com/). Ein Support-Mitarbeiter wird sich daraufhin mit Ihnen in Kontakt setzen und Sie um die Übermittelung der Artikeldaten bitten.

Die Daten Ihres Online-Shops können Sie über "Artikel > Artikel exportieren > Pangora" im Pangora-Format exportieren, indem Sie als Dateiformat "Pangora" auswählen und Ihre Merchant-ID eingeben. Sie erhalten eine XML-Datei mit den benötigten Angaben. Zusätzlich ist ein Kategorien-Mapping notwendig, d.h. Kategorien in CP::Shop müssen Kategorien von Pangora zugeordnet werden. Sie werden hier mehrere Möglichkeiten genannt bekommen.

CP::Shop stellt die Informationen über Shop-Kategorien auch per Webservices bereit. Sie können diese in einem XML-Format unter "webservices.php?file=shop&sectioninfo=all&order=+number" abrufen.

Die Artikeldaten können manuell exportiert und über ein Web-Interface bzw. per FTP an Pangora übergeben werden. Alternativ können Sie Pangora diese URL mitteilen, unter der die Artikeldaten regelmäßig automatisiert heruntergeladen werden können:

Artikeldaten im XML-Format: */index.php?file=shop&mode=export&clean=yes&type=pangora

5.6.14. Artikel exportieren zu Preisauskunft.de

Auf der Webseite http://www.preisauskunft.de/ können Online-Shopper rund um die Uhr gezielt Waren und dazugehörige Informationen finden. Über einen Link zum Anbieter kann direkt bestellt werden. Die Artikel aus CP::Shop können manuell über "Artikel > Artikel exportieren > Alternatives Format: Preisauskunft.de" im von Preisauskunft.de benötigten CSV-Format exportiert werden, indem Sie Preisauskunft.de als Alternativ-Format auswählen. Es erscheint ein Download-Fenster mit der generierten Datei. Damit Ihre Artikel bei Preisauskunft.de gelistet werden, müssen Sie sich dort einmalig anmelden.

Auch ein automatischer Export im CSV-Format ist verfügbar unter:

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=preisauskunft.de

5.6.15. Artikel exportieren zu Preisroboter.de



Preisroboter.de ist eine Suchmaschine, die die angemeldeten Online-Shops selbständig durchsucht, alle Artikelangebote des jeweiligen Shops automatisch findet und in einer Datenbank speichert. Die Artikel aus CP::Shop können manuell über "Artikel > Artikel exportieren > Alternatives Format: Preisroboter" im von Preisroboter.de benötigten CSV-Format exportiert werden, indem Sie Preisroboter.de als Alternativ-Format auswählen. Es erscheint ein Download-Fenster mit der generierten Datei.

Damit Preisroboter.de allerdings auch alle Updates automatisch in die Datenbank übernimmt, kann der automatische Export genutzt werden. Standardmäßig sind die Daten unter folgenden URL's abrufbar:

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=preisroboter.de
Informationen über Shop-Betreiber: */index.php?file=shop&mode=export&clean=yes&type=preisroboter.de&form[info]=1

Sie müssen sich jetzt nur noch unter http://www.preisroboter.de/ anmelden (bei der Anmeldung für CP::Shop als Scanmodus "preisroboter.txt / preisroboter.php" auswählen) und Ihre Shop-Daten eingeben. Für Ihren Shop öffnet sich so eine weitere interessante Plattform.

5.6.16. Artikel exportieren zu Preissuchmaschine.de



Konsumfreudige, Qualiltäts- und preisbewusste Internetnutzer mit Affinität zum Onlineshopping. PreisSuchmaschine.de-Nutzer sind gern auf dem aktuellen Stand der Technik und nutzen die vielseitigen Informationsmöglichkeiten (Testberichte, Neuheiten) aufmerksam.

Die Artikel aus CP::Shop können manuell über "Artikel > Artikel exportieren > Alternatives Format: Preissuchmaschine.de" im von Preissuchmaschine.de benötigten CSV-Format exportiert werden, indem Sie Preissuchmaschine.de als Alternativ-Format auswählen. Es erscheint ein Download-Fenster mit der generierten Datei. Damit Preissuchmaschine.de allerdings auch alle Updates automatisch in die Datenbank übernimmt, kann der automatische Export genutzt werden. Standardmäßig sind die Daten unter folgenden URL's abrufbar:

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=preissuchmaschine.de

Sie müssen sich jetzt nur noch unter http://www.preissuchmaschine.de/ anmelden und Ihre Shop-Daten eingeben. Für Ihren Shop öffnet sich so eine weitere interessante Plattform.

5.6.17. Artikel exportieren zu Preiswa(h)l.de

Über den Export zu preiswal.de können Sie sich ein weiteres Shopping-Portal für Ihre Produkte erschließen. preiswal.de konzentriert sich dabei auf Consumer-Artikel aus den Bereichen Elektronik & Photo, Computer und TV & DVD. Damit Ihre Artikel bei "www.preiswal.de" gelistet werden, müssen Sie sich dort einmalig anmelden. Ein Export im benötigten Format ist unter diesem Link möglich (bzw. über "Artikel > Artikel exportieren > Alternatives Format: Preiswal"):

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=preiswal

5.6.18. Artikel exportieren zu Schnaeppchenjagd.de



Schnaeppchenjagd.de wurde im August 1998 gegründet und legte bis heute ein rasantes Wachstum vor. Online-Händler können über den Eintrag ihrer Top-Angebote in den Schnaeppchenjagd.de - Rubriken zusätzliche Kunden-Frequenz bekommen. Die Artikel aus CP::Shop können manuell über "Artikel > Artikel exportieren > Alternatives Format: Schnaeppchenjagd.de" im von Schnaeppchenjagd.de benötigten CSV-Format exportiert werden, indem Sie Schnaeppchenjagd.de als Alternativ-Format auswählen. Es erscheint ein Download-Fenster mit der generierten Datei.

Damit Ihre Artikel bei Schnaeppchenjagd.de gelistet werden, müssen Sie sich dort einmalig anmelden. Auch ein automatischer Export ist verfügbar unter:

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=schnaeppchenjagd.de

Beachten Sie bitte auch die speziellen Preise, die wir für Sie ausgehandelt haben.
Sie erhalten HIER auch weitere Informationen zu Schnaeppchenjagd.de!

5.6.19. Artikel exportieren zu Shoppingscout24.de



ShoppingScout24 ist Teil der Scout24-Gruppe, dem führenden europäischen Netzwerk von Online-Marktplätzen, deren Angebote monatlich rund 6 Millionen Menschen nutzen. Der Export zu www.shoppingscout24.de gestaltet sich ein wenig anders als bei den anderen unterstützten Shopping-Portalen. Shoppingscout24.de benötigt kein besonderes Format in dem die Artikeldaten aufbereitet sein müssen. Stattdessen wird direkt die Suchfunktion eines Shop-Systems genutzt. Im Falle von CP::Shop ist dies die URL:

*/index.php?file=shop&mode=search&q=[SUCHBEGRIFF]&form[section]=all

Dort muss "[SUCHBEGRIFF]" immer durch den aktuellen Suchbegriff ersetzt werden. Um in die Datenbank von Shoppingscout24.de aufgenommen zu werden, müssen Sie sich also lediglich mit dem Support in Kontakt setzen und diesem Ihre URL zur Suche mitteilen (inkl. Domainname und Verzeichnis).

5.6.20. Artikel exportieren zu Shopzilla.de



Shopzilla wurde 1996 gegründet und ist heute eines der führenden Unternehmen im Bereich Comparison-Shopping im Internet. Die im kalifornischen Los Angeles angesiedelte Firma hat es sich zum Ziel gesetzt, Konsumenten behilflich zu sein, auf einfach und schnelle Weise Produkte jeglicher Art zu vergleichen und zu erwerben, unabhängig vom Anbieter und Standort.

Damit Ihre Artikel bei "www.shopzilla.de" gelistet werden, müssen Sie sich dort einmalig anmelden. Sie können die Dateien unter "Artikel > Artikel exportieren > Alternatives Format: Shopzilla" selbst exportieren und bei Shopzilla hochladen oder diese automatisch austauschen lassen:

Artikeldaten im CSV-Format: */index.php?file=shop&mode=export&clean=yes&type=shopzilla

5.6.21. Artikel exportieren zu Yatego



Yatego ist ein Online-Markplatz, auf welchem eine Vielzahl verschiedener Anbieter Ihre Artikel zum Verkauf anbieten. Die Produkte aller Anbieter befinden sich direkt in Yatego. Mit nur einem "Klick" können Artikel verschiedener Anbieter dann einfach und sicher online bestellt werden.

Damit Ihre Artikel bei "www.yatego.com" gelistet werden, müssen Sie sich dort einmalig anmelden. Im nächsten Schritt muss eine Zuordnung der Shop-Kategorien zu den Yatego-Kategorien stattfinden. Dazu gibt es bei Yatego eine gesonderte Excel-Datei, welche alle Kategorien und deren ID's auflistet. In der Datei "/modules/shop/export/articles/yatego.php" sind dann die einzelnen Zuordnung einzutragen.

Die CSV-Datei für Yatego erhalten Sie dann über "Artikel > Artikel exportieren > Alternatives Format: Yatego" oder aber Sie benutzen die folgende URL:

Artikeldaten im CSV-Forma: */index.php?file=shop&mode=export&clean=yes&type=yatego

5.6.22. Artikel hinzufügen / bearbeiten

Über die Formulare zum Hinzufügen und Bearbeiten von Artikel pflegen Sie den wichtigsten Bestandteil Ihres Online-Shops: die Artikel. Die hier angegebenen Daten werden u.a. in den Vorlagen "details.html" und "list.html" im Ordner "/modules/shop/templates" verwendet.

Diese Formulare sind selbsterklärend und erfordern genaue Angaben und Beschreibungen zu den einzelnen Artikeln (beispielsweise Titel, Preis, Steuersatz). Darüberhinaus sind zahlreiche optionale Angaben möglich wie z.B. die Hersteller-Artikelnummer und eine EAN/UPC-Nummer. Zwingend notwendig ist für jeden Artikel allerdings die Zuordnung von mindestens einer Kategorie und jeweils einem Hersteller. Nur dann kann ein Artikel auch erstellt und angezeigt werden.

Im Formular zum Bearbeiten von Artikeln stehen zusätzliche Eingabefelder und Formulare zur Verfügung. Diese dienen u.a. der Erstellung von Staffelpreisen. Auch kann ein Artikel hier einer Artikelklasse zugeordnet und mit weiteren Attributen versehen werden (d.h. spezielle Angaben bei Musik-CD's, elektrischen Geräten usw.).

Wenn Sie häufig ähnliche Artikel verkaufen, sollten Sie sich mit dem Button "Speicherung als neuer Artikel" vertraut machen. Ausgehend von einem vorhandenen Artikel erstellen Sie so einfach einen neuen Artikel ohne alle Angaben manuell machen zu müssen.

5.6.23. Artikelklassen und -attribute



Neben den normalen Formularen für Artikel bietet CP::Shop darüberhinaus weitreichende Möglichkeiten zur Individualisierung der Angaben. Dies geschieht über die sogenannten Artikelklassen, die über "Artikel > Artikelklassen" bearbeitet werden können.

Artikelklassen können z.B. Artikel wie Musik CD's oder Telefone sein. Für eine DVD beispielsweise können über die Artikelklassen einfach weitere Formularfelder für FSK, Sprache, Erscheinungsdatum, Laufzeit usw. angelegt werden. Sobald eine Artikelklasse vorhanden ist, kann diese einem existierenden Artikel zugewiesen werden. Anschließend erscheinen unterhalb des normalen Formulares die neuen Formularfelder, welche sofort mit Daten gefüllt werden können. Auf die Artikelvarianten vererbt sich die neue Artikelklasse samt Attributen ebenfalls.

Um die nun ausgefüllten Attribute auch im öffentlichen Bereich auf der Detailseite des Artikels sichtbar zu machen, müssen Sie in der genutzten Vorlage den Tag "{cp:shop:item:attributes}" einfügen. CP::Shop fügt daraufhin die Attribute ein. Die Gestaltung dieser Attribute wird per CSS vorgenommen. Nicht ausgefüllte Attibute werden automatisch versteckt.

5.6.24. Artikelvarianten

Über die Artikelvarianten können Sie Ihren Shop mit ähnlichen Artikeln befüllen, ohne bspw. Auflistungen oder Suchergebnisse für den Nutzer zu unübersichtlich zu gestalten. Artikelvarianten können immer nur über die Detailansicht des Basis-Artikels erreicht werden, werden allerdings auch im Warenkorb, in Rechnungen etc. dargestellt.

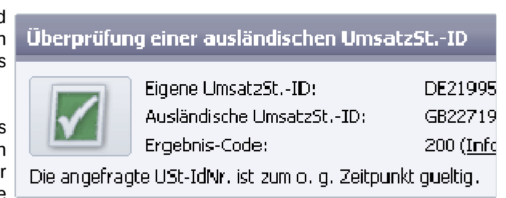
Wenn Sie eine neue Artikelvariante hinzufügen, werden automatisch die Daten des Basis-Artikels übernommen, welche dann entsprechend geändert werden. Neben den üblichen Angaben wie Titel, Beschreibung etc. stehen für die Varianten

ebenfalls die Attribute der Artikelklasse zur Verfügung, sofern eine Artikelklasse beim Basis-Artikel ausgewählt wurde.

Beachten sollten Sie, dass der Titel einer Artikelvariante automatisch immer folgendermaßen zusammengesetzt wird: "[TITEL DES BASIS-ARTIKELS]: [TITEL DER VARIANTE]". Um die Artikelvarianten auf der Webseite anzuzeigen, beachten Sie bitte die Hinweise unter "Container: Artikelvarianten anzeigen".

5.6.25. Ausländische UmsatzSt.-ID's überprüfen

In verschiedenen Bereichen von CP::Shop - wie z.B. dem Partnerprogramm - arbeiten Sie als Shop-Betreiber oft mit Unternehmen zusammen, die Sie bislang nicht kannten. Um dennoch die Seriösität der Unternehmen zu prüfen, gibt es im CP::Shop die Möglichkeit, ausländische UmsatzSt.-ID's zu überprüfen.



Im Partner-Bereich können Sie hierzu direkt neben dem Feld für die UmsatzSt.-ID über den Button "Überprüfen" ein Popup öffnen, welches Ihnen das entsprechende Ergebnis zurückliefert.

Dieser Dienst basiert auf den offiziellen Daten des Bundesamtes für Finanzen und steht, außer bei notwendigen Wartungsarbeiten, täglich zwischen 05:00 Uhr und 23:00 Uhr zur Verfügung. Je nach Auslastung kann die Abfrage zwischen wenigen Sekunden und 2-3 Minuten dauern.

Die Ihnen im Rahmen des § 6a Abs. 4 UStG obliegende Sorgfaltspflicht kann durch eine einfache Bestätigungsanfrage allerdings nicht ausreichend erfüllt werden. Eine qualifizierte Anfrage mit Angaben zu Name, Ort, Postleitzahl und Strasse ist hier nicht möglich. Solche Anfragen richten Sie bitte an das Bundesamt für Finanzen, Aussenstelle Saarlouis.

5.6.26. Bestellverwaltung / Versandbestätigung

Über die Versandbestätigung der Bestellverwaltung können Sie Ihre Kunden direkt per Email informieren, sobald eine Bestellung versandt wurde. Dies kann nützlich sein, um Ihre Kunden über den Stand der Bestellung auf dem Laufenden zu halten. Die Vorlage für diese Email können Sie dabei unter "Einstellungen > Weitere Email-Vorlagen" bearbeiten. Sie können dort u.a. die folgenden Tags verwenden (abgesehen von den normalen {cp:tags}):

{cp:shop:start}	Beginn der Anrede ("Sehr geehrter Herr..." / "Sehr geehrte Dame...")
{cp:shop:client:}*	Details zum Kunden (siehe unten)

Bei den Details zum Kunden ist für "*" eines der folgenden Wörter einzusetzen: "number", "forename", "nickname", "street", "plz", "city", "country", "company", "email", "phone", "mobile", "fax", "website", "gender" und "birthday". An die Stelle der Tags wird jeweils die entsprechende Information gestellt, z.B. "Fritz Mustermann" für "{cp:shop:client:forename} {cp:shop:client:nickname}".

In dieser Email bietet es sich an, auf weitere interessante Artikel wie z.B. Sonderaktionen hinzuweisen. Informationen über die hierfür notwendigen {cp:tags} erhalten Sie in dieser Anleitung.

5.6.27. Container: Am besten bewertete Artikel im Shop anzeigen

Über diesen Container können die von den Kunden am besten bewerteten Artikel in einer beliebigen Form im Online-Shop angezeigt werden. Der Container rechnet dazu alle Bewertungen der Kunden zusammen und ordnet die Artikel dann absteigend nach dem Durchschnittswert. Der Aufruf für diesen Container kann z.B. folgendermaßen ausschauen:

```
{cp:container module="shop" type="best_reviews" start="0" limit="5" template="best_reviews.html"}
// Anzeige der 5 am besten bewerteten Artikel über die Vorlage "best_reviews.html".
```

Die Parameter "start", "limit" und "tempalte" sind dabei optional. Über den Parameter "template" kann eine Vorlage für die Darstellung dieser Artikel ausgewählt werden. Der Pfad wird immer ausgehend vom Hauptordner angegeben. Fehlt dieser Parameter, so erfolgt die Darstellung über die normale Vorlage für die Listenansicht von Shop-Artikeln (Vorlage "/modules/shop/templates/list.html").

Parameter für diesen Container	
limit	Maximale Anzahl der Artikel (standardmäßig 10 Artikel)
start	Anzeige ab Artikel "n"
template	Vorlage für die Gestaltung der Artikel

5.6.28. Container: Anzeige der Kategorien, in welchen ein Artikel enthalten ist

Diesen Artikel finden Sie in den folgenden Kate

PC-Spiele > Englische Originalversionen

PSone > Fun- & Gesellschaftsspiele

PSone > Platinum-Serie

Über diesen Container können Sie den Besuchern zeigen, in welchen Kategorien ein bestimmter Artikel enthalten ist. Der Einsatz des Containers bietet sich dabei gerade bei sehr umfangreichen Online-Shops mit vielen Dutzend oder gar mehreren Hundert Kategorien an, da es oftmals vergleichbare Artikel auch in anderen Bereichen der Kategoriestruktur geben kann, auf die der Kunde ohne eine solche Navigation sonst nicht aufmerksam gemacht werden würde.

In einem Online-Shop für Musik-CD's könnten die einzelnen Artikel z.B. in verschiedenen Genre-Kategorien und gleichzeitig in Bestseller-Listen ("Top-Hits 2006...") gelistet sein. Auf einen Blick kann der Kunde über diesen Container dann erkennen, wo er vielleicht noch weitere für ihn interessante Artikel finden kann.

Dieser Container sollte in der Detailvorlage eines Artikels eingebunden werden. Standardmäßig ist dies die Vorlage "details.html" im Ordner "/modules/shop/templates/". In diesem Fall schaut der Containeraufruf folgendermaßen aus:

```
<strong>Diesen Artikel finden Sie in den folgenden Kategorien:</strong><br />
{cp:container module="shop" type="category_list" number="{cp:variable:number}"}
```

Prinzipiell kann der Container aber auch in jeder anderen Vorlage des Systems und in allen Bereichen genutzt werden, in denen {cp:tags} verarbeitet werden. Dann müsste statt des Tags "{cp:variable:number}" jedoch entweder die ID des gewünschten Artikels fest integriert werden oder aber ID wird dem Container über einen anderen Tag mitgeteilt.

Die Darstellung der Backtracking-Navigation kann leicht per CSS verändert werden, wenn Sie den Containeraufruf in einen <div>-Layer einschließen und diesem eine CSS-Klasse zuweisen.

5.6.29. Container: Anzeige der zuletzt aufgerufenen Artikel

Um den Besuchern auch nach dem Verlassen der Detailseite eines Artikels weiterhin diesen Artikel gezielt zum Kauf anzubieten, kann über diesen Container im Shop eine Liste der zuletzt aufgerufenen Artikel dargestellt werden. Damit diese Informationen erfasst werden können, wird mit jedem Aufruf einer Detailseite in der Session des Besuchers ein Eintrag mit der ID des Artikels hinterlassen (bis zu max. 10 Einträge). Über den Container werden diese Einträge dann ausgelesen:

```
{cp:container module="shop" type="lastvisited" [limit="5"]}
```

Die Darstellung erfolgt wie gewohnt auf der Basis von Vorlagen, wobei hier die gleichen Tags wie bei den anderen Detail- und Listenansichten zum Einsatz kommen. Über den optionalen Parameter "limit" kann zusätzlich noch die maximale Anzahl der auszugebenden Artikel festgelegt werden. Sollte der Besucher bereits mehr Artikel aufgerufen haben, so werden dann nur die max. n zuletzt aufgerufenen Artikel angezeigt.

5.6.30. Container: Anzeige des Warenkorbs auf der Webseite

Neben dem direkten Aufruf des Warenkorbs über "index.php?file=shop&mode=cart" bzw. "shop-cart.html" kann dem Shop-Kunden über einen Container der aktuelle Inhalt des Warenkorbs an einer beliebigen Position im Layout des Online-Shops angezeigt werden. Optimal eignen sich für diesen Container natürlich die Vorlagen "site.html" aus dem Ordner "/modules/shop/templates/" bzw. die Vorlage "main.htm".

```
{cp:container module="shop" type="cart"}
```

Über den obenstehenden Tag wird eine Auflistung der aktuell im Warenkorb liegenden Artikel mit Menge, Titel und jeweiligem Preis eingeblendet. Darüberhinaus wird ein Gesamtpreis sichtbar. Sollen zusätzlich die Artikelbilder eingeblendet werden, so kann der Parameter "showimages" mit dem Wert "1" hinzugefügt werden.

Alternativ stehen die folgenden Containeraufrufe zur Verfügung, um nur einzelne Informationen einzublenden (diese können dann beliebig in die Vorlagen integriert werden, beispielsweise auch zusammen mit Warenkorb-Grafiken und Textlinks):

```
{cp:container module="shop" type="cart_numtotal"}
// blendet die Anzahl der momentan im Warenkorb befindlichen Artikel ein
```

```
{cp:container module="shop" type="cart_numunique"}
// blendet die Anzahl der momentan im Warenkorb befindlichen unterschiedl. Artikel ein
```

```
{cp:container module="shop" type="cart_price"}
// blendet den Gesamtpreis (brutto) ein
```

5.6.31. Container: Artikel einblenden

Sowohl einzelne Artikel als auch komplette Auflistungen von Artikeln aus einzelnen Kategorien etc. lassen sich außerhalb des Shops auf beliebigen Seiten anzeigen und über Vorlagen darstellen. Zum Einsatz kommen hier die Container auf Basis der {cp:tags}.

Wichtig für die Einblendung eines einzelnen Artikels ist die Angabe des Parameters "number". Ist dieser vorhanden, wird nur dieser Artikel dargestellt. Ohne den Parameter "number" wird eine komplette Liste generiert.

```
{cp:container module="shop" number="9" template="sonderangebot.html"}
// Darstellung des Artikels mit der ID über die Vorlage "sonderangebot.html"
```

Jedoch können Sie auch diese Liste in der Reihenfolge noch anpassen, indem Sie über den Parameter "type" die Artikel nach verkaufter Anzahl (mit dem Wert "bestsellerbynumber") oder dem erzielten Umsatz (mit dem Wert "bestsellerbyvalue") ordnen lassen. Optionale Parameter zur Anpassung der Ausgabe der Container sind folgende: "start" (Startwert für eine Auflistung), "limit" (maximale Anzahl an Artikeln), "section" (ID der Kategorie), "order" (ordnen nach bestimmtem Feld), "dir" (Ordnungsrichtung), "template" (genutzte Vorlage) und "format" (Format der Ausgabe: HTML oder Text).

Für jeden der optionalen Parameter existieren Standardwerte, die genutzt werden, sobald der Parameter nicht angegeben wird. Bspw. wird standardmäßig aufsteigend nach der ID sortiert ohne Limitierung nach Anzahl oder Kategorie. Zum Einsatz kommt ohne Angabe von "template" die Vorlage "modules/shop/templates/list.html" (die Standardvorlage für Artikellisten).

```
{cp:container module="shop" section="12" order="title" dir="asc"}
// Artikel in Kategorie "12" aufsteigend nach dem Titel geordnet
```

```
{cp:container module="shop" type="bestseller" start="1" limit="5"}
// Einblendung der 5 am häufigsten verkauften Artikel
```

```
{cp:container module="shop" type="bestsellerbyvalue" section="4"}
// Einblendung der 5 umsatzstärksten Artikel in einer einzelnen Kategorie
```

5.6.32. Container: Artikelvarianten anzeigen

Wie in "Artikelvarianten" angesprochen, geschieht die Einbindung der Varianten eines Artikels innerhalb der Detailseite per Container. D.h. die genutzte Vorlage muss einen Container dieser Form beeinhalten:

```
{cp:container module="shop" type="variants" number="{cp:variable:number}" [template="vorlage.html"]
[order="number"] [dir="asc"]}
```

Über "{cp:variable:number}" wird dabei festgelegt, dass die richtigen Varianten angezeigt werden. Der Parameter "number" wird zwingend benötigt, die Parameter "template", "order" und "dir" hingegen sind optional und mit Standardwerten belegt.

So wird als Standardvorlage die Datei "/modules/shop/templates/details.html" genutzt und aufsteigend nach dem Titel geordnet. Für den Parameter "order" sind prinzipiell alle Spalten der Datenbanktabelle "*_shop_items" denkbar, z.B. "number", "itemnr", "title", "price", "hits" oder "date_add". Der Parameter "dir" schließlich kann als Werte "asc" oder "desc" beinhalten.

Denkbar ist über diesen Container sowohl die Darstellung der Varianten als Liste innerhalb der Detailseiten wie auch als Dropdown. Ebenso ist es möglich, dynamisch auf der Detailseite einen Link zusammenzusetzen ? la "index.php?static=datei.html&number={cp:variable:number}". Innerhalb der Datei "datei.html" könnte dann ebenfalls der Container eingesetzt werden. Somit erhalten Sie die Auflistung auf einer eigenen Seite.

5.6.33. Container: Artikelverknüpfungen (Cross-Selling / Upsells etc.) einblenden

Die einem Artikel per Artikelverknüpfungen zugeordneten Artikel werden über den Container mit dem Namen "crossselling" auf der Webseite eingeblendet. Dem Container werden dazu mindestens eine Vorlage über den Parameter "template" (ausgehend vom Hauptverzeichnis, z.B. "modules/shop/templates/list.html" für die allgemeine Listenvorlage) und die ID des gewünschten Artikels über den Parameter "number" übergeben:

```
{cp:container module="shop" type="crossselling" number="20" limit="4" template="crossselling.html"}
// Anzeige der Cross-Selling-Artikel des Artikels mit der ID 20 über die Vorlage "crossselling.html" (max. 4 Artikel).
```

Optional kann die Ausgabe über die Parameter "order", "dir" und "limit" beeinflusst werden. Je nach Gestaltung der genutzten Vorlage lassen sich so verschiedenste horizontale / vertikale Auflistungen erzeugen, u.a. mit Titel und Bild der Artikel, die auf die einzelnen Detailseiten verlinken. Über den Parameter "relation" schließlich können Sie die Auswahl auf einen bestimmten Verknüpfungstypen reduzieren, also z.B. auf höherwertige Artikel (sogenannte "Upsells").

Parameter für diesen Container	
dir	Aufsteigend ("asc") oder absteigend ("desc") ordnen
limit	Maximale Anzahl der Artikel (standardmäßig 5 Artikel)
number	ID des Artikels, dessen Cross-Selling-Artikel angezeigt werden sollen
order	Sortierung der Artikel (z.B. "title", "random", "price", "hits")
relation	Typ der Verknüpfung (1 = Cross-Selling / Zubehör, 2 = Ähnliche Artikel, 3 = Höherwertige Artikel (Upsells), 4 = Bundle-Angebote)
template	Vorlage für die Gestaltung der Cross-Selling-Artikel

5.6.34. Container: Auswahlmenü für Kategorien

Neben den normalen Navigationsmenüs für die Kategorien des Shops kann es sinnvoll sein, wenn Sie an geeigneten Stellen wie beispielsweise im Fuss-Bereich des Shops ein zusätzliches Menü zum schnellen Wechsel zwischen den Kategorien anbringen.

Ein solches Auswahlmenü in Form eines Dropdowns mit automatischem Seitenwechsel bei Auswahl kann direkt über einen Container eingebettet werden. Als Parameter sind bei diesem Container "title" (für den Text der ersten Zeile) und "category" (Startkategorie, standardmäßig "0") zulässig.

```
{cp:container module="shop" type="category_box"}
```

5.6.35. Container: Bewertung durch den Shop-Betreiber

Zusätzlich zu der Bewertungsmöglichkeit für die Kunden des Shops gibt es auch für den Shop-Betreiber die Möglichkeit, Artikel nach verschiedenen Kriterien zu bewerten. Das Bewertungs-Formular kann im Administrations-Bereich über die Seite "Artikel bearbeiten > Bewertung durch den Shop-Betreiber" aufgerufen werden. Neben einer Bewertung zwischen 0 und 5 Punkten kann zu jedem Kriterium noch ein Kommentar hinzugefügt werden.

Die Darstellung auf der Webseite (vorzugsweise in der Detailansicht eines Artikels) erfolgt wiederum über einen Container, diesmal vom Typ "owner_ratings":

```
{cp:container module="shop" type="owner_ratings" number="3" template="bewertung-betreiber.html"}
// Zeigt die Bewertung durch den Shop-Betreiber für Artikel mit der ID an.
```

Zwingend notwendig ist für diesen Container die Angabe von "number" (für den gewünschten Artikel), sowie von "template" (für die Vorlage), damit die Bewertungen auch dargestellt werden können. Fehlt der Parameter "template", so wird automatisch auf die Vorlage "/modules/shop/templates/container/owner_rating.html" zugegriffen. Innerhalb der Vorlage können {cp:tags} im Format "{cp:shop:owner_ratings:KRITERIUM:NAME}" genutzt werden:

Mögliche Werte für "KRITERIUM" (nur Standard-Kriterien)	
durability	Haltbarkeit
equipment	Ausstattung
price-performance	Preis-Leistungs-Verhältnis
quality	Qualität
total	Gesamtwertung

Mögliche Werte für "NAME"	
comment	Kommentar des Shop-Betreibers
label	Name des Kriteriums (Z.B. "Preis/Leistung")
rating	Bewertung des Shop-Betreibers

Nach obenstehenden Tabellen zeigt der {cp:tag} "{cp:shop:owner_ratings:equipment:rating}" also die Bewertung für das Kriterium "Haltbarkeit" an und ein {cp:tag} "{cp:shop:owner_ratings:total:comment}" gibt einen Kommentar für die Gesamtwertung aus. Über die zur Darstellung genutzte Vorlage können die {cp:tags} frei positioniert und gestaltet werden. Beispielsweise ist so auch die Visualisierung der Bewertung über Sternchen usw. möglich.

{cp:tags} für die Vorlagen des Containers	
{cp:shop:owner_ratings::comment}*	Kommentar des Shop-Betreibers.
{cp:shop:owner_ratings::label}*	Bezeichnung des Attributes (z.B. "price" für das Preis-Leistungs-Verhältnis)
{cp:shop:owner_ratings::rating}*	Bewertung des Shop-Betreibers.

5.6.36. Container: Eigene Artikellisten anzeigen

Wenn neben der Auflistung von Artikeln einer Kategorie oder einzelner Artikel auch eigene Artikellisten angezeigt werden sollen, dann ist dieser Listen-Container zu verwenden. Über diesen können Sie beispielsweise eine Liste mit Artikeln generieren, die Sie momentan in einer Sonderaktion (z.B. "Topten-Bücher", "Ausgezeichnete Weine 2003") anbieten oder die thematisch zusammengehörig sind.

Damit eine solche Artikelliste angezeigt werden kann, muss dem Container über den Parameter "show" eine kommaseparierte Liste der ID's der gewünschten Artikel übergeben werden. Optional kann über den Parameter "template" noch die Vorlage festgelegt werden. Standardmäßig wird als Vorlage die Datei "/modules/shop/templates/list.html" genutzt.

```
{cp:container module="shop" type="list" show="1,5,12" template="sonderangebot.html"}
// Zeigt die Artikel mit den ID's 1, 5 und 12 über die Vorlage "sonderangebot.html" an.
```

5.6.37. Container: Herstellerlisten und -details anzeigen

Über diesen Container können Sie Übersichtslisten der im Shop vertretenen Hersteller anzeigen oder aber auch Informationen eines einzelnen Herstellers. Sinnvolle Verwendungsmöglichkeiten für den Container sind u.a. die Anreicherung der Detailseite eines Artikels um Firmenname und Logo des Hersteller oder eine Übersichtsseite im Shop, auf der die gesamten Hersteller / Lieferanten den Kunden präsentiert werden. Wie immer wird diese Flexibilität durch Parameter beim Containeraufruf und die Nutzung unterschiedlicher Vorlagen erreicht.

```
{cp:container module="shop" type="companies"} // Standardanzeige
{cp:container module="shop" type="companies" limit="5" order="title"} // max. 5 Hersteller nach Titel geordnet
{cp:container module="shop" type="companies" number="1" template="company.html"} // Hersteller mit ID "1" und
Vorlage
```

Als Parameter werden - wie oben ersichtlich - bei diesem Container "template" (Pfad zur Vorlage), "order" (Ordnung), "dir" (Ordnungsrichtung), "limit" (max. Anzahl) und "start" (Startwert) unterstützt. Über den zusätzlichen Parameter "number" können Sie zudem die Ausgabe gezielt auf einen einzelnen Hersteller einschränken. Wird der Parameter "template" weggelassen, verwendet CP::Shop automatisch die Vorlage "/modules/shop/templates/container/companies.html".

Innerhalb der Vorlagen können alle Inhalte der Datenbanktabelle "*_shop_companies" über Tags im Format "{cp:shop:companies:*}" eingesetzt werden. "*" ist dabei stets durch den Namen der Tabellenspalte zu ersetzen (bspw. "{cp:shop:companies:title}" für den Firmennamen).

{cp:tags} für die Vorlage des Containers	
{cp:shop:companies:}*	Alle Angaben zum Hersteller aus der Datenbanktabelle *_shop_companies.

5.6.38. Container: Häufige Suchbegriffe als Tagcloud darstellen

Besonders häufig von Ihren Besuchern im Shop gesuchte Begriffe können in Form einer "Tagcloud" dargestellt werden. Bei einer Tagcloud handelt es sich um eine zweidimensionale Darstellung von Wörtern, bei denen wichtigere Wörte (in diesem Fall die Suchbegriffe) größer dargestellt werden, als nur selten gefragte Begriffe. Neue Besucher erhalten damit einen schnellen Überblick über die im Shop angebotenen und stark nachgefragten Artikel.

Die Tagcloud können Sie über einen Container in einer beliebigen Vorlage einbauen:

```
{cp:container module="shop" type="tag_cloud" limit="50" order="random"}
// Anzeige der Tagcloud für die 50 am häufigsten gesuchten Begriffe (zufällig geordnet).
```

Eine Vorlage wird für diesen Container nicht benötigt. Stattdessen kann die Darstellung der Tagcloud per CSS gesteuert werden. Verantwortlich sind dafür die CSS-Klassen "style_tagcloud" und "style_tagcloud_*" in den CSS-Dateien Ihres Vorlagensets. Beim Klick auf einen der Begriffe wird der Besucher automatisch auf die passenden Suchergebnisse weitergeleitet.

Parameter für diesen Container	
dir	Aufsteigend ("asc") oder absteigend ("desc") ordnen
limit	Maximale Anzahl der Suchbegriffe (standardmäßig 100 Suchbegriffe)
order	Sortierung der Suchbegriffe (z.B. "title", "random", "hits")

5.6.39. Container: Interaktive Attribute für die Übermittlung von Kundenwünschen

Neben den durch die Redakteure / den Shop-Betreiber pflegbaren Attributen einer Artikelklasse, gibt es im CP::Shop mit den sogenannten "Interaktiven Attributen" noch eine zweite Form der Attribute. Während die Angaben der normalen Attribute einer Artikelklasse innerhalb des Shops als erweiterte Informationen zu einem Artikel dargestellt werden (z.B. Informationen zur Rebsorte und zum Herkunftsgebiet eines Weines mit der Artikelklasse "Weine"), werden die Interaktiven Attribute direkt durch die Kunden gepflegt.

Dies bedeutet, dass Kunden bei Artikeln mit derartigen Attributen zusätzliche Angaben hinterlassen, die beispielsweise zur Personalisierung eines Artikels genutzt werden können. Ein Beispiel soll hier den praktischen Einsatz der Interaktiven Attribute verdeutlichen:

In einem Online-Shop für Weine werden spezielle Geschenksets zu einem festen Preis zum Kauf angeboten. Die Kunden sollen dabei jeweils max. 3 Weine aus einer Liste auswählen können. Weiterhin soll das Geschenkset in verschiedenen Farbkombinationen / Stilen verfügbar sein (z.B. "Sommer", "Winter") und die Kunden sollen einen zusätzlichen Text für eine Grußkarte eingeben können.

Mit dem Einsatz der Interaktiven Attribute ist für die Realisierung dieses Beispieles nun keine zusätzliche Programmierung mehr erforderlich. Es müssen lediglich die folgenden Schritte durchgeführt werden:

1. Es wird eine neue Artikelklasse angelegt mit den Feldern, die später durch die Kunden ausgefüllt werden sollen. In diesem Beispiel wären dies ein Feld vom Typ "Textbereich (interaktiv)" für den Text der Grußkarte sowie insgesamt 4 Felder vom Typ "Dropdownfeld (interaktiv)" für die Auswahl der Weine sowie der Farbkombination.
2. Die gewünschten Artikel werden der neuen Artikelklasse zugeordnet.
3. In die für die Darstellung der Detailansicht der Artikel verwendeten Vorlage (standardmäßig ist dies die Vorlage "/modules/shop/templates/details.html) muss der im Folgenden gezeigte {cp:tag} eingebaut werden.

Dieser Container sorgt für die Anzeige des Formulares für die Interaktiven Attribute:

```
{cp:container module="shop" type="interactive_attributes" number="{cp:variable:number}"}
```

In der Detailansicht der Artikel werden dann durch die Kunden die einzelnen Felder ausgefüllt. Die dort gemachten Angaben werden intern gespeichert und im Anschluss im Warenkorb angezeigt. Ebenso werden diese Angaben in der PDF-, HTML- und Text-Version der Rechnung mit angezeigt. Eine komplett individuelle Anpassung der bestellten Artikel wird mit Hilfe der Interaktiven Attributen also nun erstmals möglich.

5.6.40. Container: Kategorien / Kategoriestruktur anzeigen

Wie in Contentpapst und CP::Shop üblich, werden Navigationsmenüs auf der Basis der {cp:tags} und Container aufgebaut. Hierbei lässt sich das einfachste Menü über diesen Tag fertig einbauen:

```
{cp:container module="shop" type="categories"}
```

Doch dies ist natürlich noch nicht der volle Funktionsumfang des Containers. CP::Shop nutzt für das Menü standardmäßig die Vorlage "/modules/shop/templates/container/sections.html". Über den Parameter "template" kann aber eine alternative Vorlage aufgerufen werden, welche dann beliebig gestaltet werden kann. Und die optionalen Parameter "category" (Oberkategorie, standardmäßig "0"), "order" (Ordnung der Kategorien, standardmäßig "position"), "dir" (Ordnungsrichtung, "asc" oder "desc"), "limit" (max. Anzahl), "start" (Startwert) und "num_cols" (Anzahl der Spalten) stehen ebenfalls zur Verfügung.

Wenn nur die Details einer einzelnen Kategorie angezeigt werden sollen, so kann statt der oben beschriebenen Parameter auch direkt der Parameter "number" für die ID der anzuzeigenden Kategorie genutzt werden.

Besonders interessant (gerade für Sitemaps) ist die Möglichkeit, innerhalb der verwendeten Vorlage erneut einen Container einzubetten. Dort muss dann der Parameter "category" über den Tag "{cp:shop:number}" aufgerufen werden, damit die nächst tiefere Ebene angezeigt wird.

5.6.41. Container: Kundenmeinungen

Zu den Artikeln im CP::Shop können Kunden und Besucher ihre individuelle Meinung mit Kommentar samt einer Bewertung auf einer Skala von 0 bis 5 hinterlassen. Wie mit diesen Kundenmeinungen / Rezensionen verfahren wird, ist abhängig von Ihrer Ausrichtung als Shop-Betreiber:

- Kundenmeinungen können direkt unterhalb eines Artikels oder auf einer Extraseite angezeigt werden
- Kundenmeinungen können auch nur intern angezeigt werden (quasi als Feedback für Sie als Shop-Betreiber)

Die vorhandenen Kundenmeinungen eines Artikels blenden Sie auf der Detailseite (standardmäßig über die Vorlage "/modules/shop/templates/details.html") mit diesem Containeraufruf ein:

```
{cp:container module="shop" type="reviews" number="{cp:variable:number}" limit="5"}
```

Der Parameter "limit" bestimmt hierbei die maximale Anzahl an Kundenmeinungen, über "page" kann die Seite festgelegt werden und mit "template" können Si eine alternative Vorlage festlegen. Wenn "template" nicht vorhanden ist, wird zur Darstellung die Vorlage unter "/modules/shop/templates/container/reviews.html" genutzt.

{cp:tags} für die Vorlagen des Containers	
{cp:shop:review:}*	Alle Angaben zur Kundenmeinung aus der Datenbanktabelle *_shop_reviews.
{cp:shop:reviews:item_name}	Name des Artikels, zu dem die Meinung abgegeben wurde.
{cp:shop:reviews:item_permalink}	Permalinks des Artikels, zu dem die Meinung abgegeben wurde.
{cp:shop:reviews:total}	Anzahl der insgesamt bzw. zu einem Artikel verfügbaren Kundenmeinungen.
{cp:shop:reviews:rating}	Durchschnittliche Bewertung des Artikels durch die Kunden.

Um nicht nur die vorhandenen Kundenmeinungen anzuzeigen, sondern den Kunden die Möglichkeit zu geben, eigene Kundenmeinungen hinzuzufügen, muss folgender Containeraufruf integriert werden:

```
{cp:container module="shop" type="review_box"}
```

Das Formular für die Eingabe einer neuen Kundenmeinung wird über die Vorlage "/modules/shop/templates/container/review_box.html" dargestellt. Über "CP::Shop > Einstellungen > Kundenmeinungen" schließlich können Sie die Einstellungen für die Kundenmeinungen anpassen.

Und durch die Erweiterung der Vorlagen um PHP ist z.B. eine Beschränkung der Kundenmeinungen auf eingeloggte Kunden denkbar. Beachten Sie hierzu die Anleitungen in unserer Knowledge Base auf http://www.sandoba.de/.

5.6.42. Container: Mediendateien eines Artikels einblenden

Um die einem Artikel zugeordneten Mediendateien wie Pressemeldungen, Produkttests oder Screenshots auch auf der Webseite anzuzeigen, wird ein Container vom Typ "media" genutzt. Dem Container müssen zunächst über den Parameter "number" die ID des Artikels und über "template" die zu nutzende Vorlage mitgeteilt werden, alle weiteren Parameter sind optional:

```
{cp:container module="shop" type="media" number="5" template="mediendateien_archive.html"
filetype="sit,pdf,ace,rar"}
// Mediendateien in den Archiv-Formaten für den Artikel "5" über die Vorlage "mediendateien_archive.html" anzeigen
```

Anstelle einer festen ID kann im Parameter "number" natürlich auch ein {cp:tag} eingesetzt werden. Erforderlich ist dies z.B., wenn die Mediendateien direkt innerhalb der Detailansicht eines Artikels genutzt werden sollen. Dann muss in diesem Parameter der {cp:tag} "{cp:variable:number}" eingesetzt werden. Das System entnimmt die ID des Artikels dann direkt dem Parameter "number" in der URL. Somit werden bei jedem Artikel nur die jeweils passenden Mediendateien angezeigt. Anhand der angegbenen Vorlage werden die Mediendateien dann auf der Webseite dargestellt. Dabei können in der Vorlage die unten aufgeführten {cp:tags} für die Inhalte eingesetzt werden. Zusätzlich besteht die Möglichkeit, über den Parameter "filetype" gezielt nur bestimmte Dateitypen ausgeben zu lassen (mit "gif,jpg" bspw. nur Grafiken im GIF- und JPG-Format für eine Galerie usw.).

Eine Vorlage für die Mediendateien ist standardmäßig noch nicht vorhanden und muss neu angelegt werden, z.B. als "media.html" im Ordner "/modules/shop/templates/container/". Der Aufruf in der Detailansicht der Artikel sieht dann folgendermaßen aus:

```
{cp:container module="shop" type="media" number="{cp:variable:number}"
template="modules/shop/templates/container/media.html"}
```

Der Parameter "order" erlaubt die Sortierung der Mediendateien nach den Werten "number" (ID), "title" (Titel), "url" (Dateiname) und "filetype" (Dateityp). Die Ordnungsrichtung wird durch den Parameter "dir" mit den Werten "asc" für aufsteigend und "desc" für absteigend bestimmt. Werden diese Parameter nicht mit angegeben, dann werden die Mediendateien alphabetisch in aufsteigender Reihenfolge angezeigt.

{cp:tags} für die Vorlagen des Containers	
{cp:shop:media:}*	Alle Angaben zur Mediendatei aus der Datenbanktabelle *_shop_media.
{cp:shop:media:description}	Beschreibung zu dieser Mediendatei
{cp:shop:media:filetype}	Dateityp (z.B. "exe", "pdf", "png")
{cp:shop:media:number}	Interne ID der Mediendatei
{cp:shop:media:title}	Titel der Mediendatei
{cp:shop:media:url}	Pfad zur Datei inklusive des Dateinames

5.6.43. Container: Neueste Artikel anzeigen

Um die neuesten Artikel Ihren Benutzern präsentieren zu können, müssen Sie dazu einen Container in der gewünschen Vorlage platzieren. Sobald dieser Container platziert ist, werden die im angegebenen Zeitraum hinzugefügten Artikel angezeigt. Die neuesten Artikel stehen dabei am Anfang.

```
{cp:container module="shop" type="last_articles" [days="3"] [template="vorlage.html"]}
```

Die Parameter "days", "limit" und "template" sind optional und mit Standardwerten belegt. Für "days", also den max.

Zeitraum seit dem Hinzufügen der Artikel in Tagen, ist als Standardwert "5" gesetzt. Standardvorlage ist die Datei "/modules/shop/templates/list.html". "limit" gibt die max. Anzahl an Artikeln an. Der Standardwert für "limit" liegt bei 10.

5.6.44. Container: Preisangaben direkt in Fremdwährungen umrechnen lassen

Wenn Sie den Shop international nutzen möchten, dann werden Sie dafür ebenfalls verschiedene Währungen aktivieren müssen. Der Kunde kann dann je nach seinem Wohnort die für ihn zutreffende Währung auswählen. Zusätzlich kann es jedoch sinnvoll sein, wenn man direkt auf einer bestimmten Seite neben dem Preis in der Originalwährung gleich die Preise in den restlichen Währungen anzeigen kann. Genau dies ist mit dem "currencies"-Container möglich.

Der Aufruf der Containers in einer beliebigen Vorlage erfolgt folgendermaßen:

```
{cp:container module="shop" type="currencies" price="4,69" template="{cp:path}vorlage.html"}
```

D.h. dem Container wird der Preis übergeben (dies kann statt des festen Preises auch der entsprechende {cp:tag} z.B. von den Detailseiten eines Artikels sein) und es muss eine Vorlage angegeben werden, über welche die anderen Preise dargestellt werden. Wenn Sie eine andere Währung als den Euro als Standard-Währung verwenden, kann können Sie über den Parameter "currency" im Containeraufruf noch den Namen der aktuell aktivierten Währung angeben und die Preise werden automatisch entsprechend umgerechnet.

Die für die Anzeige genutzte Vorlage könnte z.B. den folgenden Inhalt haben, um die Preise in anderen Währungen anzuzeigen:

```
Originalpreis: {cp:shop:currency:price} EUR<br />
Preis in USD: {cp:shop:currency:USD} USD<br />
Preis in NZD: {cp:shop:currency:NZD} NZD
```

Die einzelnen Kürzel für die Währungen erhalten Sie direkt in der Shop-Verwaltung unter "Modul-Einstellungen > Währungen". Über den {cp:tag} "{cp:shop:currency:price}" wird der Originalpreis in der Vorlage angezeigt.

{cp:tags} für die Vorlagen des Containers	
{cp:shop:currency:}*	Umgerechneter Preis für das jeweilige Währungskürzel (z.B. "{cp:shop:currency:NZD}").
{cp:shop:currency:price}	Preis des Artikels in der Standardwährung.

5.6.45. Container: Preisstaffel eines Artikels anzeigen

Die für einen Artikel im Administrationsbereich festgelegte Preisstaffel wird über einen Container auf der Webseite eingeblendet. Sinnvollerweise findet diese Einblendung in der Detailansicht der Artikel statt. Dem Container muss dazu die Nummer des gewünschten Artikels übergeben werden:

```
{cp:container module="shop" type="prices" number="1240" order="amount" dir="asc"}
// Anzeige der Preisstaffel des Artikels mit der ID 1240 (aufsteigend geordnet nach der Anzahl).
```

Optional kann die Ausgabe über die Parameter "order", "dir" und "template" beeinflusst werden. Standardmäßig wird die Preisstaffel über die Vorlage "/modules/shop/templates/container/prices.html" gestaltet, über den Parameter "template" kann jedoch auch eine alternative Vorlage genutzt werden.

Parameter für diesen Container	
dir	Aufsteigend ("asc") oder absteigend ("desc") ordnen
number	ID des Artikels, dessen Preisstaffel angezeigt werden soll.
order	Sortierung der Preise (nach "amount" oder "price")
template	Vorlage für die Gestaltung der Preisstaffel

{cp:tags} für die Vorlagen des Containers	
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items.
{cp:shop:prices:amount}	Mindest-Anzahl des Artikels für diesen Level der Staffelpreise.
{cp:shop:prices:baseprice}	Basispreis des Artikels, wenn ein Gewicht in Gramm angegeben wurde.
{cp:shop:prices:currency}	Kürzel der aktuell ausgewählten Währung.
{cp:shop:prices:price}	Nettopreis für diesen Level der Staffelpreise.
{cp:shop:prices:taxprice}	Bruttopreis für diesen Level der Staffelpreise.
{cp:shop:prices:discount_price}	Netto-Rabattpreis für diesen Level der Staffelpreise gegenüber dem Standard-Preis.
{cp:shop:prices:discount_taxprice}	Brutto-Rabattpreis für diesen Level der Staffelpreise gegenüber dem Standard-Preis.
{cp:shop:prices:discount_percent}	Rabatt in Prozent gegenüber dem Standard-Preis.

5.6.46. Container: Suchbegriffe der Besucher auf der Webseite anzeigen

Im Unterschied zum "tag_cloud"-Container werden bei diesem Container die Suchbegriffe der Besucher nicht nach ihrer Häufigkeit gruppiert. Vielmehr können diese alphabetisch, nach Datum bzw. nach Anzahl der zutreffenden Artikel aus dem Shop sortiert werden. Diese Auflistung können Sie über einen Container in einer beliebigen Vorlage einbauen:

```
{cp:container module="shop" type="searches" order="date" dir="desc" limit="10"
template="themes/cpshop/search_list.html"}
// Anzeige der 10 letzten Suchbegriffen über die neue Vorlage "search_list.html" im Ordner "/themes/cpshop/".
```

Eine Vorlage wird für diesen Container zwingend benötigt, alle anderen Parameter sind optional:

Parameter für diesen Container	
dir	Aufsteigend ("asc") oder absteigend ("desc") ordnen
limit	Maximale Anzahl der Suchbegriffe (standardmäßig 100 Suchbegriffe)
order	Sortierung der Suchbegriffe (z.B. "details", "date", "hits")
start	Anzeige ab Suchbegriff "n"
template	Vorlage für die Gestaltung der Suchbegriffe

{cp:tags} für die Vorlagen des Containers	
{cp:shop:searches:date}	Datum der letzten Anfrage mit diesem Suchbegriff
{cp:shop:searches:comment}	Eingegebener Suchbegriff
{cp:shop:searches:hits}	Anzahl der zutreffenden Artikel (Suchergebnisse)

5.6.47. Container: Trend der Artikelpreise anzeigen



Für viele Ihrer Kunden (besonders die Stammkunden) dürfte es interessant sein, wie sich die Preis in Ihrem Shop verändern. Gerade bei Artikeln mit häufig wechselnden Preisen wie z.B. Computerteilen stellt deshalb ein Preistrend eine hervorragende Möglichkeit dar, Kunden auch bei einem großen Artikelsortiment stets die Artikel zu zeigen, deren Preis sich besonders stark geändert hat (Preiserhöhung und -senkung).

Auf der linken Seite sehen Sie ein Beispiel, wie dies realisiert werden kann. Der Preistrend kann dabei in einer beliebigen Vorlage, z.B. auch innerhalb eines Popups eingebaut werden. In CP::Shop ist dies - wie könnte es anders sein - über einen Container zu realisieren. Dieser hat das folgende Format (Beispiel):

```
{cp:container module="shop" type="pricetrend" section="3" limit="5" start="0" template="preistrend.html"}
```

In diesem Beispiel wird der Preistrend über die gleichnamige Vorlage "preistrend.html" dargestellt. Es werden nur max. 5 Artikel aus der Kategorie mit der ID 3 angezeigt, angefangen mit dem ersten Artikel. Nicht unbedingt notwendig sind die Parameter "section", "limit", "start" und "template". Als Standard-Vorlage wird - wenn keine andere Vorlage angegeben wurde - die Datei "/modules/shop/templates/container/pricetrend.html" genutzt. In dieser ist auch der benötigte Aufbau einer solchen Vorlage ersichtlich. Der Preistrend der Artikel wird über zwei Grafiken aus dem Verzeichnis "/modules/shop/images/" verdeutlicht.

5.6.48. Container: Versandarten detailliert beschreiben

Im Bestellprozess müssen Ihre Kunden u.a. eine Zahlungs- und eine Versandart aus den von Ihnen freigeschalteten Möglichkeiten auswählen. Für die Versandarten wird standardmäßig nur ein Auswahlfeld mit Titel und Preis angezeigt. Sie können jedoch Ihren Kunden die angebotenen Versandarten auch detaillierter beschreiben und diese auf einer eigenen Seite auflisten. Dazu lesen Sie sich zuerst in der Anleitung den Eintrag "Individuelle Texte einfügen" durch und legen Sie eine neue Datei an. Dort können einleitende Informationen usw. platziert werden.

Die Versandarten werden dann als Container eingebunden:

```
{cp:container module="shop" type="shipment" [template="modules/shop/templates/container/shipment.html"] cache=
["86400"]}
```

Die Parameter "template" und "cache" sind hierbei optional. Sofern keine Vorlage angegeben wird, verwendet CP::Shop automatisch die Vorlage aus diesem Beispiel. In der Datei "shipment.html" (bzw. Ihrer jeweiligen Vorlage) können dann die folgenden Tags verwendet werden, um die Versandart nach Ihren Vorstellungen zu präsentieren:

{cp:tags} für die Vorlagen des Containers	
{cp:shop:shipment:}*	Alle Angaben zur Versandart aus der Datenbanktabelle *_shop_shipment.
{cp:shop:shipment:currency}	Kürzel der aktuell ausgewählten Währung.
{cp:shop:shipment:description}	Beschreibung
{cp:shop:shipment:image}	Grafik (wenn nicht vorhanden, wird ein leeres Bild verwendet)
{cp:shop:shipment:price}	Kosten für diese Versandart
{cp:shop:shipment:showurl}	Sichtbare URL mit Link
{cp:shop:shipment:title}	Titel der Versandart / des Anbieters
{cp:shop:shipment:url}	URL der Versandanbieters

5.6.49. Container: Weitere Container

Neben den in dieser Anleitung detailliert beschriebenen Containern mit verschiedenen Anpassungsmöglichkeiten über die Parameter bietet CP::Shop eine Reihe weiterer Container. Diese Container dienen hauptsächlich der Einblendung von zusätzlichen Informationen wie beispielsweise der Lieferzeiten etc.:

- *add_box* - Box mit Formular zum schnellen Hinzufügen von Artikeln zum Warenkorb
- *currency_box* - blendet die im Shop verfügbaren Währungen ein (zusätzliche Parameter: "template")
- *delivery* - blendet die Lieferzeiten des Shops ein
- *login_box* - je nach Status Box für Kunden-Login oder -Konto
- *search_box* - blendet ein Formular zur Suche im Shop ein

Diese Container werden jeweils mit dem untenstehenden Tag aufgerufen, wobei "[CONTAINERNAME]" mit dem oben kursiv geschriebenen Namen des Containers ersetzt werden muss. Die Anpassung der Container kann über die Vorlagen im Ordner "/modules/shop/templates/container/" erfolgen.

```
{cp:container module="shop" type="[CONTAINERNAME]"}
```

5.6.50. Container: Zahlungsarten detailliert beschreiben

Im Bestellprozess müssen Ihre Kunden u.a. eine Zahlungs- und eine Versandart aus den von Ihnen freigeschalteten Möglichkeiten auswählen. Für die Zahlungsarten wird standardmäßig nur ein Auswahlfeld mit dem Titel angezeigt. Sie können jedoch Ihren Kunden die angebotenen Zahlungsarten auch mit allen Details vorstellen und diese auf einer eigenen Seite auflisten. Dazu lesen Sie sich zuerst in der Anleitung den Eintrag "Individuelle Texte einfügen" durch und legen Sie eine neue Datei an. Dort können einleitende Informationen usw. platziert werden. Die Zahlungsarten selbst werden als Container eingebunden:

```
{cp:container module="shop" type="payment" [template="modules/shop/templates/container/payment.html"]
[cache="86400"]}
```

Die Parameter "template" und "cache" sind hierbei optional. Sofern keine Vorlage angegeben wird, verwendet CP::Shop automatisch die Vorlage aus diesem Beispiel. In der Datei "/modules/shop/templates/container/payment.html" (bzw. Ihrer jeweiligen Vorlage) können dann die folgenden Tags verwendet werden, um die Versandart nach Ihren Vorstellungen zu präsentieren:

{cp:tags} für die Vorlagen des Containers	
{cp:shop:payment:}*	Alle Angaben zur Zahlungsart aus der Datenbanktabelle *_shop_payment.
{cp:shop:payment:currency}	Kürzel der aktuell ausgewählten Währung.
{cp:shop:payment:description}	Beschreibung
{cp:shop:payment:image}	Grafik (wenn nicht vorhanden, wird ein leeres Bild verwendet)
{cp:shop:payment:maxamount}	Maximaler Bestellwert
{cp:shop:payment:minamount}	Minimaler Bestellwert
{cp:shop:payment:min_orders}	Minimal notwendige Bestellungen
{cp:shop:payment:price}	Kosten für diese Zahlungsart
{cp:shop:payment:start_date}	Laufzeit von
{cp:shop:payment:stop_date}	Laufzeit bis
{cp:shop:payment:title}	Titel der Zahlungsart / des Anbieters

5.6.51. Container: Zufällig ausgewählte Artikel anzeigen

Per Container lassen sich - in einer beliebigen Vorlage eingesetzt - zufällig ausgewählte Artikel anzeigen. So können Sie Ihrem Besucher bspw. auf der Startseite ein stets wechselndes Angebot präsentieren. Diese Artikel können dabei aus einer einzelnen Kategorie entnommen werden oder aus dem gesamten Angebot. Stets werden allerdings nur verfügbare und freigeschaltete Artikel angezeigt.

```
{cp:container module="shop" type="random" [limit="5"] [template="vorlage.html"] [section="3"] [cache="86400"]}
```

Alle Parameter nach "type" sind bei diesem Container optional. Standardmäßig werden bis zu 5 Artikel (Parameter "limit") über die Vorlage "/modules/shop/templates/list.html" (Parameter "template") dargestellt. Wird der Parameter "section" mit der ID einer Kategorie angegeben, dann werden nur Artikel dieser Kategorie angezeigt. Über den Parameter "cache" kann wie gewohnt der Inhalt des Containers über eine beliebig lange Zeit (in Sekunden) zwischengespeichert werden.

5.6.52. Container: Zuletzt gekaufte Artikel

Sie können Ihren Besuchern weitere Kaufanreize bieten, indem Sie ihnen zeigen, welche Artikel nicht nur besonders gut laufen, sondern z.B. erst kurz zuvor verkauft wurden. In CP::Shop gibt es eigens für diesen Fall einen eigenen Container "last_order". Der Container stellt eine beliebige Anzahl an Artikeln anhand einer Vorlage dar. Über den Parameter "template" muss daher zwigend eine Vorlage angegeben werden. Der Parameter "limit" hingegen ist - ebenso wie der Parameter "start" (Startwert) - optional. Wenn "limit" nicht angegeben ist, wird nur ein Artikel ausgegeben.

```
{cp:container module="shop" type="last_order" template="vorlage.html"}
```

5.6.53. Container: Ähnliche Artikel / "Kunden, die x gekauft haben..."

Sicherlich kennen Sie aus Online-Shops die Möglichkeit, Kunden beim Aufruf eines Artikels darauf hinzuweisen, welche Artikel andere Kunden noch gekauft haben. Dieser Verweis auf ähnliche / besonders beliebte Artikel kann die Umsätze im Online-Shop zusätzlich steigern.

Auch CP::Shop verfügt über eine derartige Funktion. Wenn Sie die ähnlichen Artikel zu einem bestimmten Artikel einblenden wollen, so müssen Sie in einer beliebigen Vorlage (standardmäßig natürlich in "/modules/shop/templates/details.html" für die Detailansicht) folgenden {cp:tag} integrieren:

```
{cp:container module="shop" type="related" number="{cp:variable:number}" limit="5" shorten_shorttext="40"
format="text"}
```

Dieser Containeraufruf nutzt den über "&number=..." in der URL angesteuerten Artikel. Alternativ ist auch das direkte Setzen einer Artikel-Nummer möglich oder Sie setzen hier mehrere Artikel-Nummern ein (durch Komma getrennt).

Die Darstellung basiert auf der Vorlage "/modules/shop/templates/related.html". Natürlich haben Sie über den Parameter "template" wieder die Möglichkeit, eine alternative Vorlage zu verwenden. In der Vorlage selbst können die gleichen {cp:tags} wie in der Detailansicht eines Artikels verwendet werden. Im oben gezeigten Beispiel werden maximal 5 ähnliche Artikel angezeigt. Die Länge des Feldes "shorttext" wird auf 40 Zeichen begrenzt.

{cp:tags} für die Vorlagen des Containers	
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items.
{cp:shop:related:num_items}	Anzahl der insgesamt verfügbaren ähnlichen Artikel.

5.6.54. Einstellungen: Hersteller

Jedem Artikel in CP::Shop ist ein Hersteller zugewiesen. Das kann der tatsächliche Hersteller des Artikels sein oder auch ein Lieferant oder Großhändler. Bevor ein Artikel hinzugefügt werden kann, muss mindestens ein Hersteller vorhanden sein.

Für jeden Hersteller geben Sie dessen Kontaktdaten an. Sie haben hier auch die Möglichkeit, einen direkten Link zu Bestell-Seite sowie Ihre Kundennummer zu hinterlassen. Nachbestellungen von Artikeln sind so schnell möglich. Auf der Basis dieser Herstellerübersicht werden später im Shop Funktionen wie die Auflistung von Artikeln eines Herstellers realisiert. Sie sollten diesen Prozess also möglichst sorgfältig durchführen.

5.6.55. Einstellungen: Versandarten



Über "Einstellungen > Versandarten" können Sie die gewünschten Versandarten für Ihren Online-Shop freischalten. Es sind hier bereits verschiedene Unternehmen wie die Dt. Post, FedEx, TNT Express oder trans-o-flex voreingetragen. Zu jeder Versandart können Titel, Beschreibung, Preis, ein Link zur Webseite des Unternehmens und ein Logo angegeben werden. Weiterhin ist ein Status einzugeben und eine Versandart muss als Standard festgelegt werden.

Die freigeschalteten Versandarten sind dann im Online-Shop von Ihren Besuchern bei einer Bestellung auswählbar. Zusätzlich können diese auf einer Übersichtsseite mit Beschreibung usw. sichtbar gemacht werden (siehe hierzu auch "Versandarten detailliert beschreiben").

5.6.56. Einstellungen: Zahlungsarten



Über "Einstellungen > Zahlungsarten" können Sie die gewünschten Zahlungsarten für Ihren Online-Shop freischalten. Es sind hier bereits verschiedene Zahlungsarten wie Vorkasse, Rechnung, Nachnahme usw. voreingetragen. Weitere Zahlungsarten (Bezahlung per Kreditkarte, Telefonrechnung, Handy etc.) können optional hinzufügt werden (siehe "optionale Zahlungsarten für CP::Shop"). Zu jeder Zahlungsart können mind. Titel, Beschreibung, Preis und ein Logo angegeben werden. Weiterhin sind ein Status und optional die URL zu einem Payment-Script einzugeben. Eine Zahlungsart muss als Standard festgelegt werden.

Die freigeschalteten Zahlungsarten sind dann im Online-Shop von Ihren Besuchern direkt im Bestellprozess auswählbar. Dort können diese über die Vorlage "modules/shop/templates/cart/cart_1.html" dargestellt werden.

5.6.57. Emails / Bestellbestätigung

Für die Bestätigung einer Bestellung wird von Contentpapst automatisch nach der Bestellung eine Email an den Kunden samt Rechnung im PDF-Format gesendet, sowie je nach Einstellung die gleiche Email an den Shop-Betreiber. Die hierfür genutzte Vorlage ist die Datei "/modules/shop/templates/emails/email.order.tpl".

Neben allgemeinen Angaben zum Kunden (Anschrift) und zur Bestellung (bestellte Artikel, Preise, Gesamtpreis, Versand- und Zahlungsart), können für Zahlungsarten, die eine Überweisung erforderlich machen, zusätzliche Informationen eingeblendet werden. Hat der Kunde eine Zahlung per Überweisung/Vorkasse gewählt, dann sollten Sie ihm zusätzlich zu den normalen Angaben u.a. eine Bankverbindung übermitteln. Mittels der Konstruktion "{cp:if:shoptransfer} [BANKDATEN] {/cp:if:shoptransfer}" wird die Auswahl auf diese Kunden begrenzt. Alle notwendigen Angaben können Sie dann an der Stelle von "[BANKDATEN]" einsetzen.

Wie Sie dies optimal gestalten, sehen Sie am bestem am Beispiel der bereits fertig angepassten Vorlage "/modules/shop/templates/emails/de.order.tpl".

5.6.58. Erweiterte Informationen im Shop einbinden

Wenn Sie im Shop erweiterte Informationen wie z.B. Ihre Allgemeinen Geschäftsbedingungen, Liefer- oder Kontaktinformationen einbinden wollen, so können dafür die von CP::Shop bereitgestellten Funktionen genutzt werden. Sie benötigen dazu jeweils eine HTML-Datei mit den anzuzeigenden Inhalten, welche Sie im Ordner "/modules/shop/templates/info/" abspeichern. Wichtig: Diese Datei sollte nur den <body>-Bereich der HTML-Datei enthalten, d.h. es darf sich nicht um eine komplette HTML-Datei handeln.

Anschließend öffnen Sie die Datei "/modules/shop/public/info.php". In dieser Datei wird über ein Array festgelegt, in welchen Dateien die über "/index.php?file=shop&mode=info&type=[TYP]" abgerufenen Inhalte gespeichert sind. Sie müssen sich zunächst entscheiden, unter welchem Namen ein Abruf möglich sein soll. Eine als "agb-info.html" gespeicherte Datei könnte als Typ beispielsweise "agb" erhalten. Dann fügen Sie dort im Array den Eintrag mit Typ und Pfad hinzu. Ein Abruf ist dann sofort als "/index.php?file=shop&mode=info&type=agb" verfügbar.

5.6.59. Individuelle Texte einfügen

Im öffentlichen Bereich des Shops können Sie auch ohne das Content-Management-System einfach weitere individuelle

Texte im Shop-Layout einsetzen. Sie müssen hierfür lediglich im Ordner "/modules/shop/templates/info/" eine neue HTML-Datei anlegen. Der erste Teil des Dateinamen wird dann als Parameter in einer URL nach diesem Format eingefügt:

```
*/index.php?file=shop&mode=info&type=[DATEINAME]
```

Die Datei "shipment.html" wird dann z.B. über "*/index.php?file=shop&mode=info&type=shipment" aufgerufen und in das normale Shop-Layout eingebettet. Innerhalb dieser HTML-Dateien können wieder beliebige {cp:tags} verwendet werden.

5.6.60. Integrierte und optionale Zahlungsanbieter

CP::Shop verfügt über verschiedene Schnittstellen zu Zahlungsanbietern (Payment Provider). Diese Zahlungsanbieter ermöglichen gegen einen best. Betrag pro Transaktion (oder auch eine monatl. Pauschale) u.a. die Abwicklung von Kreditkartenzahlungen. Die Schnittstellen zu den einzelnen Anbietern werden direkt über die grafische Oberfläche von CP::Shop konfiguriert. Notwendige Daten wie die Merchant- und Shop-ID erhalten Sie als Shop-Betreiber nach der Freischaltung beim Zahlungsanbieter.

Neben den standardmäßig integrierten Zahlungsarten / -anbietern bieten wir Ihnen weitere optionale Zahlungsanbieter an (z.B. Postbank PaySolution und ExperCash). Eine Übersicht über diese Zahlungsanbieter erhalten Sie unter http://www.sandoba.de/produkte/shop-software-cpshop/zahlungsanbieter/. Weitere Schnittstellen zu Zahlungsanbietern können auch individuell erstellt werden.

5.6.61. Kategorien: Allgemein

Unter "CP::Shop > Kategorien" können Sie die Kategorien Ihres Online-Shops verwalten. Beim Aufruf sehen Sie im linken Teil der Seite ein Fenster, in welchem später der Kategorienbaum angezeigt werden wird. Sie können dort direkt die Struktur Ihrer Kategorien einsehen. In diesem Fenster haben Sie die Möglichkeit:

- die Reihenfolge der Kategorien zu ändern (wenn mehrere Kategorien auf einer Ebene vorhanden sind)
- Kategorien zu löschen (rekursiv, wenn Unterkategorien vorhanden sind)
- die Bearbeitung der Kategorien aufzurufen (durch einen Klick auf den Titel der Kategorie)

Im rechten Teil hingegen können neue Kategorien hinzugefügt, sowie bearbeitet und gelöscht werden. Das Formular zum Hinzufügen einer Kategorie bietet Ihnen Felder für Titel, Beschreibung, ein Bild, den Status und eine Auswahl für die Einordnung in die vorhandene Kategoriestruktur. Die übrigen Felder für eine Kategorie erscheinen erst bei der Bearbeitung der Kategorie. Dort steht Ihnen u.a. zusätzlich ein WYSIWYG-Editor für eine ausführliche Beschreibung der Kategorie zur Verfügung.

5.6.62. Kategorien: Anpassung der Darstellung

Auch die Kategorien des CP::Shop werden über Vorlagen dargestellt. Standardmäßig ist dies die Datei "/modules/shop/templates/section.html". Diese Vorlage zeigt u.a. Titel und Beschreibung der Kategorie an, schafft Platz für den restlichen Inhalt (Anzeige der Artikel in dieser Kategorie) und blendet die Navigation zum Wechsel auf die weiteren Seiten ein.

In der Gestaltung der Vorlage sind Sie wie immer frei und können daher an dieser Stelle bspw. Flash-Animationen einbinden oder {cp:tags} für andere Artikel etc. Im Folgenden eine Beschreibung der möglichen Tags:

{cp:shop:number}	Nummer der Kategorie (z.B. für Verlinkung benötigt)
{cp:shop:title}	Titel der Kategorie
{cp:shop:description}	Beschreibung (unformatierte Kurzbeschreibung)
{cp:shop:image}	Grafik der Kategorie (komplett als HTML-Tag, sofern die Grafik vorhanden ist)
{cp:shop:full_text}	ausführliche Beschreibung (formatiert, aus WYSIWYG-Editor)
{cp:shop:pages}	Navigation zum Seitenwechsel (z.B. "1 2 3 weiter")
{cp:shop:numitems}	Anzahl der Artikel in dieser Kategorie
{cp:shop:showitems}	Artikel, die momentan angezeigt werden (z.B. "1 - 5 von 6 Artikeln")
{cp:content}	Hier wird der restl. Inhalt eingesetzt (unbedingt notwendig)

Der Tag {cp:shop:showitems} kann dabei nur auf den Einzelseiten einer Kategorie und nicht auf der Übersichtseite genutzt werden (da hier stets nur die ersten Artikel gezeigt werden).

5.6.63. Neukennzeichnung für n-Tage

Beim Import bzw. dem manuellen Hinzufügen neuer Artikel wird automatisch im Hintergrund das Einstellungsdatum abgespeichert. Um den Besuchern Ihres Online-Shops die neuen Artikel auffälliger zu präsentieren, können Sie diese für einen bestimmten Zeitraum mit einer passenden Grafik versehen. Hierzu stellen Sie in den Optionen unter "Neukennzeichnung für n-Tage" den gewünschten Zeitraum in Tagen ein.

Ist ein Artikel jünger als der angegebene Zeitraum, so wird die Grafik unter "/themes/[IHR VORLAGEN-SET]/images/new.gif" genutzt. Um die Grafik auch auf der Webseite sichtbar zu machen, fügen Sie in die Vorlagen der Detail- oder Listenansicht bzw. eine Container-Vorlage eines Artikels den Tag "{cp:shop:item:is_new}" ein. Bei älteren Artikeln wird der Tag leergelassen.

5.6.64. Partner: Tracking von Affiliate-Programmen

Über die im Shop integrierten Funktionen lässt sich sehr gezielt der Erfolg von Affiliate-Programmen und Werbemassnahmen (Google Adwords, Overture etc.) beobachten. Zum einen gibt es hier natürlich die allgemeine Statistik mit der Refererauswertung. Diese erlaubt allerdings lediglich die Auswertung von Clicks etc. Wenn Sie hingegen die Anzahl

der Verkäufe messen wollen, so haben Sie im Wesentlichen zwei Möglichkeiten:

- der Werbepartner wird als neuer Partner in CP::Shop angelegt und entsprechend verlinkt
- alle über den Werbepartner gewonnenen Kunden werden automatisch in einer Gruppe zusammengefasst

Durch die Kombination beider Möglichkeiten zur Messung der Erfolges können Sie noch genauere Aussagen treffen. Bei der ersten Möglichkeit werden alle Bestellungen die über den Partner durchgeführt werden anteilig für diesen vergütet. Hierbei können Sie natürlich die Provision auf "0" stellen, da der Partner ja nicht aktiv am Partnerprogramm teilnimmt. Eine Verlinkung auf den Shop kann dann bspw. über "index.php?file=shop&partner=google" durchgeführt werden.

Für die zweite Möglichkeit müssen Sie Links wie z.B. "index.php?file=shop&grp=1" nutzen. Hierbei wird in der Variable "grp" die ID der Kundengruppe festgehalten, in welche die Kunden automatisch kommen, sobald Sie über einen solchen Link bestellen. Wenn Sie eine Gruppe "Google-Adwords" anlegen, dann können Sie so stets genau beobachten, wie erfolgreich die dortige Werbeschaltung tatsächlich ist.

5.6.65. Staffelpreise für Artikel festlegen

Für jeden Artikel können gesonderte Staffelpreise festgelegt werden. Der Preis eines einzelnen Artikels verringert sich damit für den Kunden bei Erhöhung der bestellten Anzahl. Die Angabe von Staffelpreisen ist immer optional. Ebenso ist die Anzahl der Ebenen nicht beschränkt.

Staffelpreise sind bei der Bearbeitung eines Artikels unter "Preisstaffelung für diesen Artikel" anzugeben. Über den Button "Alle entfernen" werden alle Staffelpreise gelöscht, nur der normale Preis wird weiterhin berechnet. Alternativ können einzelne Staffelpreise entfernt werden, indem das Feld für Preis oder Anzahl freigelassen wird. Alle Staffelpreise werden nach der Anzahl geordnet aufgelistet.

5.6.66. Statistiken

Bereits integriert in CP::Shop ist ein umfangreiches Statistiktool. Neben den allgemeinen Statistiken von Contentpapst können Sie hier genau ablesen, welche Artikel besonders beliebt sind, welche Versand- und Zahlungsarten häufig gewählt werden sowie allgemeine Angaben wie die Anzahl der Kunden und die Höhe des Umsatzes.

Darüberhinaus wird aus den einzelnen Angaben auch berechnet, welche Kunden besonders häufig bzw. besonders viel bei Ihnen gekauft haben. Diesen können Sie beispielsweise besondere Rabattstufen zuweisen.

Bevor Sie einen Einblick in die Statistiken erhalten, müssen Sie zuerst einen neuen Statistik-Report auswählen. Sie müssen hierzu

- den gewünschten Zeitraum (fester bzw. variabler Zeitraum) und
- die benötigten Auswertungen

festlegen. Anschließend geben Sie dem Statistik-Report einen Titel. Nachdem dieser gespeichert wurde, können Sie den Statistik-Report direkt aufrufen. Folgende Auswertungen stehen für jeden Statistik-Report zur Auswahl:

- Artikel im Warenkorb
- Bestellte Artikel nach Verkaufsrang
- Bestellte Artikel nach Umsatz
- Bestellte Artikel nach Hersteller
- Bestellte Artikel nach Hersteller (detailliert)
- Conversion Rate (Verhältnis Besucher:Kunden)
- Bestellungen pro Stunde
- Bestellungen (in den letzten 60 Tagen)
- Kunden mit meisten Bestellungen
- Kunden mit höchstem Umsatz
- Artikelansichten (ohne Zeiteingrenzung)
- Kategorieansichten (ohne Zeiteingrenzung)
- Favoritisierte Versandarten
- Favoritisierte Zahlungsarten
- Suchwort-Statistik
- Partner mit meisten Bestellungen
- Partner mit höchstem Umsatz

5.6.67. Suchmöglichkeiten im Shop

Im öffentlichen Bereich des Shops ist bereits eine Standard-Suchfunktion integriert, die Sie über "index.php?file=shop&mode=search" bzw. "shop-seach.html" aufrufen können. Dort lassen sich Suchbegriffe eintragen, wobei der Besucher die Möglichkeit hat, die Suchergebnisse zu sortieren (nach Titel, Preis und den bisherigen Aufrufen) und auf einzelne Kategorien zu begrenzen. Alle über diese Suchfunktion getätigten Abfragen werden intern mitgeloggt und können später im Administrationsbereich in der Statistik eingesehen werden.

5.6.68. Umstellung von Online-Shop auf Katalog

CP::Shop lässt sich durch den flexiblen Aufbau über Vorlagen auch einfach in einen Katalog mit einzelnen Artikeln umwandeln. Besonders für Auto- und Immobilienhändler ist nicht immer der volle Funktionsumfang mit Online-Bestellung, Partnerprogramm, Webservices, Gutscheinen usw. notwendig oder die Artikel sind so komplex oder beratungsbedürftig, dass ein individueller Austausch zwischen Händler und Kunden erfolgen muss.

Alle nicht benötigten Funktionen können aus dem öffentlichen Bereich des Shops entfernt werden, indem die Vorlagen im Ordner "/modules/shop/templates/" bearbeitet werden. Die Vorlage "details.html" in diesem Ordner beispielsweise dient zur Darstellung der Detailansicht eines Artikels. Hier können Sie statt dem Link zur Bestellung auch ein Kontaktformular für den Besucher o.ä. integrieren.

5.6.69. Vorlage "cart/cart_0.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/cart/" und dient der Anzeige des Warenkorbs.

{cp:tags} für diese Vorlage	
{cp:shop:attributes}	Werte der ausgewählten Interaktiven Attribute eines Artikels (innerhalb der Artikelauflistung).
{cp:shop:edit}	Buttons zum Ändern der Anzahl bzw. zum Entfernen des Artikels aus dem Warenkorb (innerhalb der Artikelauflistung).
{cp:shop:editnumber}	ID eines Warenkorb-Eintrags (innerhalb der Artikelauflistung).
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items (innerhalb der Artikelauflistung).
{cp:shop:item:currency}	Kürzel der aktuell ausgewählten Währung (innerhalb der Artikelauflistung).
{cp:shop:price:final}	Brutto-Gesamtpreis aller momentan im Warenkorb liegenden Artikel.
{cp:shop:price:price}	Bruttopreis des Artikels für die entsprechende Anzahl der im Warenkorb liegenden Einheiten (innerhalb der Artikelauflistung).
{cp:shop:value}	Anzahl der momentan im Warenkorb liegenden Einheiten des Artikels (innerhalb der Artikelauflistung).

5.6.70. Vorlage "cart/cart_1.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/cart/" und dient der Anzeige der zweiten Seite im Bestellprozess (Login bzw. Registrierung als Neukunde).

{cp:tags} für diese Vorlage	
{cp:shop:country}	Liste mit den Ländern des Geschäfts-/Liefergebietes des Online-Shops, Standard-Land ist vorausgewählt.
{cp:shop:fav_countries}	5 Länder des Geschäfts-/Liefergebietes in absteigender Reihenfolge, aus denen die meisten Kunden stammen

5.6.71. Vorlage "cart/cart_2.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/cart/" und dient der Anzeige der dritten Seite im Bestellprozess (Auswahl der Zahlungs- und Versandart).

{cp:tags} für diese Vorlage	
{cp:shop:}*	Alle Angaben zur Zahlungsart aus der Datenbanktabelle *_shop_payment.
{cp:shop:checked}	Zahlungsart ausgewählt ("1") oder nicht ausgewählt ("0").
{cp:shop:client:}*	Alle Angaben zu einem eingeloggten Kunden aus der Datenbanktabelle *_shop_clients.
{cp:shop:currency}	Kürzel der aktuell ausgewählten Währung.
{cp:shop:form}	Weitere Eingabefelder zu einer Zahlungsart.
{cp:shop:payment:num_payment}	Anzahl der für den Kunden verfügbaren Zahlungsarten.
{cp:shop:shipment}	Auswahlliste der Versandarten im dritten Schritt des Bestellprozesses für Neukunden.
{cp:shop:shipment:message}	Meldung, wenn keine Versandart für diesen Kunden verfügbar ist.

5.6.72. Vorlage "cart/cart_3.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/cart/" und dient der Anzeige der letzten Seite des Bestellprozesses (Übersicht und Bestätigung der gemachten Angaben).

{cp:tags} für diese Vorlage	
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items (innerhalb der Artikelauflistung).
{cp:shop:item:currency}	Kürzel der aktuell ausgewählten Währung (innerhalb der Artikelauflistung).
{cp:shop:price:price}	Bruttopreis des Artikels (innerhalb der Artikelauflistung).
{cp:shop:value}	Anzahl der Einheiten des Artikels (innerhalb der Artikelauflistung).
{cp:shop:payment}	Angaben zur gewählten Zahlungsart.
{cp:shop:payment:value}	ID der gewählten Zahlungsart.
{cp:shop:shipment}	Angaben zur gewählten Versandart.
{cp:shop:shipment:value}	ID der gewählten Versandart.
{cp:shop:price:coupon}	Höhe des Rabattes durch einen eventuell eingelösten Gutschein.
{cp:shop:price:final}	Bruttogesamtpreis der Bestellung des Kunden.
{cp:shop:price:payment}	Preis der gewählten Zahlungsart.
{cp:shop:price:shipment}	Preis der gewählten Versandart.
{cp:shop:price:total}	Bruttozwischenpreis der Bestellung (ohne Versand- und Zahlungsart, ohne eventuellen Gutschein).

5.6.73. Vorlage "cart/cart_express.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/cart/" und dient der Express-Bestellung eines vorhandenen Kunden im Shop.

{cp:tags} für diese Vorlage	
{cp:shop:payment}	Auswahlformular für die Zahlungsarten, Standard-Zahlungsart vorausgewählt.
{cp:shop:shipment}	Auswahlformular für die Versandarten, Standard-Versandart vorausgewählt.

5.6.74. Vorlage "cart/cart_sendto.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/cart/" und dient der Anzeige des Formulares für eine zusätzliche Versandanschrift.

{cp:tags} für diese Vorlage	
{cp:shop:country}	Auswahlformular für die Länder, Standard-Land ist vorausgewählt.

5.6.75. Vorlage "client/account.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/client/" und dient der Anzeige der Startseite des Kundenbereiches.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zum Kunden aus der Datenbanktabelle *_shop_clients.
{cp:shop:order:}*	Alle Angaben zur Bestellung aus der Datenbanktabelle *_shop_order.
{cp:shop:order:order_number}	Bestellnummer (Format z.B. "1207-9291").
{cp:shop:order:statusimage}	Bild zum aktuellen Status der Bestellung.
{cp:shop:order:statustext}	Text zum aktuellen Status der Bestellung.
{cp:shop:num_order}	Anzahl der bisherigen Bestellungen des Kunden.

5.6.76. Vorlage "client/edit_address.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/client/" und dient der Bearbeitung der zusätzlichen Lieferadressen im Kunden-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:address:}*	Alle Angaben zum Kunden an dieser Lieferadresse aus der Datenbanktabelle *_shop_clients.
{cp:shop:address:delete}	Button, um eine Lieferadresse wieder löschen zu können.
{cp:shop:address:gender}	Anrede des Kunden an dieser Lieferadresse ("Herr", "Frau").
{cp:shop:client:country}	Auswahlformular für die Länder, aktuelles Land ist vorausgewählt.
{cp:shop:client:gender}	Auswahlformular für die Anrede ("Herr", "Frau").

5.6.77. Vorlage "client/edit_order.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/client/" und dient der Bearbeitung von Bestellungen im Kunden-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zur Rechnungsadresse des Kunden aus der Datenbanktabelle *_shop_clients.
{cp:shop:item:currency}	Kürzel der aktuell ausgewählten Währung.
{cp:shop:order:}*	Alle Angaben zur Bestellung des Kunden aus der Datenbanktabelle *_shop_order.
{cp:shop:order:order_number}	Bestellnummer (Format z.B. "1207-9291").
{cp:shop:order:pdf}	URL für den Aufruf der Rechnung im PDF-Format.
{cp:shop:order:price}	Gesamtbruttopreis der Bestellung.
{cp:shop:order:order_itemnumber}	ID des Artikels innerhalb der Bestellung aus der Datenbanktabelle *_shop_order_items.
{cp:shop:sendto:}*	Alle Angaben zur Lieferadresse des Kunden aus der Datenbanktabelle *_shop_clients.

5.6.78. Vorlage "client/edit_payment.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/client/" und dient der Bearbeitung der Zahlungseinstellungen im Kunden-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zum Kunden aus der Datenbanktabelle *_shop_clients.

5.6.79. Vorlage "client/edit_wishlist.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/" und dient der Bearbeitung des Wunschzettels im Kunden-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items.
{cp:shop:num_wishlist}	Anzahl der Wunschzettel eines Kunden.
{cp:shop:num_wishlist_items}	Anzahl der Artikel auf dem Wunschzettel eines Kunden.
{cp:shop:wishlist:}*	Alle Angaben zu den Artikeln des Wunschzettel aus der Datenbanktabelle *_shop_wishlist_items.

5.6.80. Vorlage "client/options.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/" und dient der Bearbeitung der allgemeinen Angaben im

Kunden-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zum Kunden aus der Datenbanktabelle *_shop_clients.
{cp:shop:client:country}	Auswahlformular für die Länder, aktuelles Land ist vorausgewählt.
{cp:shop:client:gender}	Geschlecht des Kunden (bzw. "Herr", "Frau").

5.6.81. Vorlage "container / review_box.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/container/" und dient der Abgabe einer Kundenmeinungen zu einem Artikel.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zum Kunden aus der Datenbanktabelle *_shop_clients.

5.6.82. Vorlage "container / search.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/container/" und dient der Anzeige eines Suchformulares.

{cp:tags} für diese Vorlage	
{cp:shop:search:sections}	Auswahlliste für die Kategorien in der Schnellsuche des Shops.

5.6.83. Vorlage "emails / de.availability.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Information des Shop-Betreibers über Artikel mit geringem Warenbestand.

{cp:tags} für diese Vorlage	
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items.
{cp:shop:item:url}	URL für die Bearbeitung des Artikels im Administrationsbereich.

5.6.84. Vorlage "emails / de.cancellation.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Information des Shop-Betreibers über Stornierungen von Artikeln / Bestellungen.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zum Kunden aus der Datenbanktabelle *_shop_clients.
{cp:shop:order:id}	ID der Bestellung.
{cp:shop:order:order_number}	Bestellnummer (Format z.B. "1207-9291").

5.6.85. Vorlage "emails / de.companies.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Information eines Herstellers / Lieferanten über einen ausverkauften Artikel, so dass dieser nachbestellt werden kann.

{cp:tags} für diese Vorlage	
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items.

5.6.86. Vorlage "emails / de.order.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Anzeige der Bestellbestätigungsemail.

{cp:tags} für diese Vorlage	
{cp:shop:attributes}	Interaktive Attribute des Artikels (innerhalb der Artikelauflistung).
{cp:shop:client:}*	Alle Angaben zum Kunden aus der Datenbanktabelle *_shop_clients.
{cp:shop:client:name}	Vollständiger Name des Kunden (Vor- und Nachname zusammengesetzt).
{cp:shop:comment}	Bei der Bestellung durch den Kunden eingegebener Kommentar.
{cp:shop:count}	Anzahl der Einheiten eines Artikels (innerhalb der Artikelauflistung).
{cp:shop:currency}	Kürzel der aktuell ausgewählten Währung.
{cp:shop:deliverytime}	Lieferzeit des Artikels (innerhalb der Artikelauflistung).
{cp:shop:itemcomment}	Ausgewählte Interaktive Attribute zu einem Artikel (innerhalb der Artikelauflistung).
{cp:shop:itemnr}	Artikelnummer des Artikels (innerhalb der Artikelauflistung).
{cp:shop:number}	ID des Artikels (innerhalb der Artikelauflistung).
{cp:shop:order:}*	Alle Angaben zur Bestellung aus der Datenbanktabelle *_shop_order.
{cp:shop:order_number}	Bestellnummer (Format z.B. "1207-9291").
{cp:shop:payment}	Name der gewählten Zahlungsart.
{cp:shop:price}	Bruttopreis des Artikels (innerhalb der Artikelauflistung).

{cp:shop:price_coupon}	Höhe des Rabattes durch einen eventuell eingelösten Gutschein.
{cp:shop:price_final}	Bruttogesamtpreis der Bestellung des Kunden.
{cp:shop:price_payment}	Preis der gewählten Zahlungsart.
{cp:shop:price_shipment}	Preis der gewählten Versandart.
{cp:shop:shipment}	Name der gewählten Versandart.
{cp:shop:start}	Ansprache des Kunden, abhängig vom Geschlecht.
{cp:shop:sendto:}*	Alle Angaben zur Lieferadresse des Kunden aus der Datenbanktabelle *_shop_clients.
{cp:shop:sendto:name}	Vollständiger Name zur Lieferadresse des Kunden (Vor- und Nachname zusammengesetzt).
{cp:shop:taxcomment}	Abhängig von der Kundengruppe des Kunden Vermerk "inkl. MwSt." oder "ohne MwSt.".
{cp:shop:title}	Titel des Artikels (innerhalb der Artikelauflistung).

5.6.87. Vorlage "emails/de.partner_free_mail.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Information des neuen Partners über die Möglichkeiten des Partnerprogrammes nach seiner Freischaltung.

{cp:tags} für diese Vorlage	
{cp:shop:partner:}*	Alle Angaben zum Partner aus der Datenbanktabelle *_shop_partner.

5.6.88. Vorlage "emails/de.partner_message.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient dem Versand einer Meldung des Partners an den Shop-Betreiber.

{cp:tags} für diese Vorlage	
{cp:shop:partner:}*	Alle Angaben zum Partner aus der Datenbanktabelle *_shop_partner.
{cp:shop:partner:message}	Nachricht des Partners

5.6.89. Vorlage "emails/de.partner_recovery.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und versorgt einen Partner mit einem neuen Passwort für sein Partner-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:partner:}*	Alle Angaben zum Partner aus der Datenbanktabelle *_shop_partner.
{cp:shop:partner:password}	Neu generiertes Passwort des Partners in unverschlüsselter Form.

5.6.90. Vorlage "emails/de.partner_register.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Information des Shop-Betreibers über eine neue Partner-Anmeldung.

{cp:tags} für diese Vorlage	
{cp:shop:partner:}*	Alle Angaben zum Partner aus der Datenbanktabelle *_shop_partner.

5.6.91. Vorlage "emails/de.priceinfo.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Information des Shop-Betreibers über eine Wunschpreis-Anforderung eines Besuchers.

{cp:tags} für diese Vorlage	
{cp:shop:priceinfo:}*	Im Formular für die Anforderung eines Wunschpreises eingegebene Daten.

5.6.92. Vorlage "emails/de.recommend.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Weiterempfehlung eines Artikels durch einen Besucher / Kunden.

{cp:tags} für diese Vorlage	
{cp:shop:comment}	Eingebener Kommentar des Besuchers.
{cp:shop:from_email}	Email-Adresse des Absenders.
{cp:shop:from_name}	Name des Absendes.
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items.
{cp:shop:item:currency}	Kürzel der aktuell ausgewählten Währung.
{cp:shop:price:taxprice}	Brutto-Preis des empfohlenen Artikels.
{cp:shop:to_email}	Name des Empfängers.
{cp:shop:to_name}	Email-Adresse des Empfängers.

5.6.93. Vorlage "emails/de.recovery.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und versorgt einen Kunden mit einem neuen Passwort für sein Kunden-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zum Kunden aus der Datenbanktabelle *_shop_clients.
{cp:shop:client:password}	Neu generiertes Passwort des Kunden in unverschlüsselter Form.

5.6.94. Vorlage "emails/de.register.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und versorgt den gerade neu angemeldeten Kunden u.a. mit seinem Passwort für das Kunden-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zum Kunden aus der Datenbanktabelle *_shop_clients.
{cp:shop:client:country}	Herkunftsland des Kunden.
{cp:shop:client:name}	Vollständiger Name des Kunden (Vor- und Nachname zusammengesetzt).
{cp:shop:client:password}	Bei der Anmeldung generiertes Passwort in unverschlüsselter Form.
{cp:shop:start}	Ansprache des Kunden, abhängig vom Geschlecht.

5.6.95. Vorlage "emails/de.register_admin.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Information des Shop-Betreibers über eine neue Kunden-Anmeldung.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zum Kunden aus der Datenbanktabelle *_shop_clients.

5.6.96. Vorlage "emails/de.return.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient dem Versand einer RMA-/Rückgabe-Anforderung eines Kunden.

{cp:tags} für diese Vorlage	
{cp:shop:return:clientnumber}	Kundennummer
{cp:shop:return:comment}	Grund der Rückgabe
{cp:shop:return:email}	Email-Adresse
{cp:shop:return:itemnumber}	Artikelnummer
{cp:shop:return:name}	Name
{cp:shop:return:ordernumber}	Rechnungsnummer

5.6.97. Vorlage "emails/de.reviews.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Benachrichtigung des Shop-Betreibers über neue Kundenmeinungen.

{cp:tags} für diese Vorlage	
{cp:shop:reviews:}*	Alle Angaben zur Kundenmeinung aus der Datenbanktabelle *_shop_reviews.
{cp:shop:reviews:client}	Name des Kunden
{cp:shop:reviews:description}	Beschreibung der Kundenmeinung
{cp:shop:reviews:item}	Titel des bewerteten Artikels
{cp:shop:reviews:rating}	Bewertung des Artikels
{cp:shop:reviews:title}	Titel der Kundenmeinung

5.6.98. Vorlage "emails/de.shipment.tpl"

Diese Vorlage liegt im Ordner "/modules/shop/templates/emails/" und dient der Anzeige der Versandbestätigungsemail.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zum Kunden aus der Datenbanktabelle *_shop_clients.
{cp:shop:order:}*	Alle Angaben zur Bestellung aus der Datenbanktabelle *_shop_order.
{cp:shop:order:tracking}	Paket-Tracking-Nummer.
{cp:shop:order:tracking_name}	Name des Versanddienstleisters für das Paket-Tracking.
{cp:shop:order:tracking_url}	URL des Versanddienstleisters für das Paket-Tracking.
{cp:shop:start}	Ansprache des Kunden, abhängig vom Geschlecht.

5.6.99. Vorlage "export_html.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/" und dient dem HTML-Export von Artikeln.

{cp:tags} für diese Vorlage	
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items.
{cp:shop:item:currency}	Kürzel der aktuell ausgewählten Währung.
{cp:shop:item:price}	Nettopreis des Artikels.
{cp:shop:item:section}	Name der ersten zugeordneten Kategorie.
{cp:shop:item:taxprice}	Bruttopreis des Artikels.
{cp:shop:item:url}	URL für den Aufruf des Artikels im öffentlichen Bereich.

5.6.100. Vorlage "message.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/" und ist die Standardvorlage für die Ausgabe einer Meldung an den Besucher / Kunden.

{cp:tags} für diese Vorlage	
{cp:shop:message}	Textmeldung, z.B. Fehlermeldung. Der Inhalt der Meldung ist variabel.
{cp:shop:order:}*	Alle Angaben zur Bestellung aus der Datenbanktabelle *_shop_order.
{cp:shop:target}	Button, der den Besucher nach dem Durchlesen der Meldung auf eine variable Zielseite bringt.

5.6.101. Vorlage "partner/ads.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/partner/" und dient der Anzeige der für die Partner verfügbaren Werbemittel.

{cp:tags} für diese Vorlage	
{cp:shop:partner:}*	Alle Angaben zum Partner aus der Datenbanktabelle *_shop_partner.

5.6.102. Vorlage "partner/options.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/partner/" und dient der Bearbeitung der allgemeinen Angaben im Partner-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:partner:}*	Alle Angaben zum Partner aus der Datenbanktabelle *_shop_partner.
{cp:shop:partner:country}	Auswahlformular für die Länder, Land des Partners ist vorausgewählt.

5.6.103. Vorlage "partner/order.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/partner/" und dient der Anzeige der bisherigen Provisionszahlungen im Partner-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:orderout:}*	Alle Angaben zu den Partner-Bestellungen aus der Datenbanktabelle *_shop_partner_orderout.
{cp:shop:comment:orderout}	Infotext zu den angefallenen Partner-Bestellungen.

5.6.104. Vorlage "partner/partner_2.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/partner/" und dient der Anzeige des Anmeldeformulares für Partner.

{cp:tags} für diese Vorlage	
{cp:shop:country}	Auswahlformular für die Länder, Standard-Land ist vorausgewählt.

5.6.105. Vorlage "partner/payment.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/partner/" und dient der Bearbeitung der Bankverbindung im Partner-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:partner:}*	Alle Angaben zum Partner aus der Datenbanktabelle *_shop_partner.

5.6.106. Vorlage "partner/statistic.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/partner/" und dient der Anzeige der (Klick-)Statistiken im Partner-Konto.

{cp:tags} für diese Vorlage	
{cp:shop:clicks:}*	Alle Angaben zu den Partner-Klicks aus der Datenbanktabelle *_shop_partner_clicks.

{cp:shop:comment:clicks}	Infotext zu den angefallenen Partner-Klicks.
{cp:shop:comment:order}	Infotext zu den angefallenen Partner-Bestellungen.
{cp:shop:order:}*	Alle Angaben zu den Partner-Bestellungen aus der Datenbanktabelle *_shop_partner_order.
{cp:shop:partner:country}	Land des Partners.
{cp:shop:partner:message}	Meldung des Partners an den Shop-Betreiber.

5.6.107. Vorlage "payment / anypay.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/payment/" und dient der Bezahlung über AnyPay nach dem Abschluss der Bestellung im Shop.

{cp:tags} für diese Vorlage	
{cp:shop:anypay_url}	URL der Zahlungsseite bei Anypay für den Aufruf in einem Popup o.ä.

5.6.108. Vorlage "payment / gzspaymaster.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/payment/" und dient der Bezahlung über GZS Paymaster nach dem Abschluss der Bestellung im Shop.

{cp:tags} für diese Vorlage	
{cp:shop:gzspaymaster_url}	URL der Zahlungsseite bei GZS Paymaster für den Aufruf in einem Popup o.ä.

5.6.109. Vorlage "payment / infin.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/payment/" und dient der Bezahlung über Infin nach dem Abschluss der Bestellung im Shop.

{cp:tags} für diese Vorlage	
{cp:shop:infin_url}	URL der Zahlungsseite bei Infin für den Aufruf in einem Popup o.ä.

5.6.110. Vorlage "payment / mpay24.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/payment/" und dient der Bezahlung über mPAY24 nach dem Abschluss der Bestellung im Shop.

{cp:tags} für diese Vorlage	
{cp:shop:mpay24_url}	URL der Zahlungsseite bei mPAY24 für den Aufruf in einem Popup o.ä.

5.6.111. Vorlage "payment / paypal.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/payment/" und dient der Bezahlung über PayPal nach dem Abschluss der Bestellung im Shop.

{cp:tags} für diese Vorlage	
{cp:shop:address}	Gesamte Rechnungsadresse des Kunden.
{cp:shop:amount}	Bruttopreis der Bestellung in EUR.
{cp:shop:currency}	Kürzel der Standard-Währung des Shops.
{cp:shop:invoice}	ID der Bestellung, für welche eine Bezahlung per PayPal vorgenommen wird.
{cp:shop:notify_url}	Info-URL für die Rückmeldung von PayPal an den Shop.
{cp:shop:paypal_email}	PayPal-Email (aus den PayPal-Einstellungen).
{cp:shop:paypal_image}	Logo bei Bestellung (aus den PayPal-Einstellungen).
{cp:shop:cancel_return}	Error-URL für die Rückmeldung von PayPal an den Shop.
{cp:shop:paypal_text}	Untertitel bei Bestellung (aus den PayPal-Einstellungen).
{cp:shop:paypal_url}	Fertige URL für das PayPal-Popup.
{cp:shop:return}	Return-URL für die Rückmeldung von PayPal an den Shop.
{cp:shop:verify}	Hash-Code zur Verifizierung der PayPal-Rückmeldung an den Shop.

5.6.112. Vorlage "payment / saferpay.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/payment/" und dient der Bezahlung über Saferpay nach dem Abschluss der Bestellung im Shop.

{cp:tags} für diese Vorlage	
{cp:shop:saferpay_url}	URL der Zahlungsseite bei Saferpay für den Aufruf in einem Popup o.ä.

5.6.113. Vorlage "payment / tpay.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/payment/" und dient der Bezahlung über T-Pay nach dem Abschluss der Bestellung im Shop.

{cp:tags} für diese Vorlage	
{cp:shop:tpay_url}	URL der Zahlungsseite bei T-Pay für den Aufruf in einem Popup o.ä.

5.6.114. Vorlage "payment/yellowpay.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/payment/" und dient der Bezahlung per Yellowpay nach dem Abschluss der Bestellung im Shop.

{cp:tags} für diese Vorlage	
{cp:shop:yellowpay_url}	URL der Zahlungsseite bei Yellowpay für den Aufruf in einem Popup o.ä.

5.6.115. Vorlage "recommend.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/" und dient der Anzeige des Formulares für die Weiterempfehlung von Artikeln.

{cp:tags} für diese Vorlage	
{cp:shop:name}	Name des Artikels, welcher weiterempfohlen werden soll.
{cp:shop:number}	ID dieses Artikels.

5.6.116. Vorlage "recommend_success.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/" und zeigt eine Meldung nach der Weiterempfehlung eines Artikels an.

{cp:tags} für diese Vorlage	
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items.

5.6.117. Vorlage "register.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/" und zeigt das Formular für die Registrierung eines Kunden an.

{cp:tags} für diese Vorlage	
{cp:shop:country}	Liste mit den Ländern des Geschäfts-/Liefergebietes des Online-Shops, Standard-Land ist vorausgewählt.
{cp:shop:fav_countries}	5 Länder des Geschäfts-/Liefergebietes in absteigender Reihenfolge, aus denen die meisten Kunden stammen

5.6.118. Vorlage "search.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/" und dient der Ausgabe der Shop-Suche.

{cp:tags} für diese Vorlage	
{cp:shop:num_items}	Anzahl der insgesamt verfügbaren Suchergebnisse.
{cp:shop:numpages}	Anzahl der insgesamt verfügbaren Seiten mit Suchergebnissen.
{cp:shop:page}	Nummer der aktuell angezeigten Suchergebnisseite.
{cp:shop:search:company}	Auswahlliste der im Shop verfügbaren Hersteller.
{cp:shop:search:currency}	Kürzel der aktuell ausgewählten Währung.
{cp:shop:search:date}	Auswahlliste für die Eingrezung der Zeit ("Hinzugefügt seit...").
{cp:shop:search:dir}	Auswahlliste der Sortierrichtungen (aufsteigend und absteigend).
{cp:shop:search:order}	Auswahlliste der im Shop verfügbaren Sortiermöglichkeiten.
{cp:shop:search:per_page}	Maximale Anzahl der Artikel pro Seite.
{cp:shop:search:price_max}	Vom Besucher eingegebener maximaler Preis.
{cp:shop:search:price_min}	Vom Besucher eingegebener minimaler Preis.
{cp:shop:search:section}	Auswahlliste der im Shop verfügbaren Kategorien.
{cp:shop:search:string}	Vom Besucher eingegebener Suchbegriff.
{cp:shop:search_url}	URL zum Anspringen einer bestimmten Seite der Suchergebnisse.
{cp:shop:showitems}	Bereich der aktuell angezeigten Suchergebnisse ("Artikel n1 bis n2 ...").

5.6.119. Vorlage "site.html"

Diese Vorlage liegt im Ordner "/modules/shop/templates/" und bildet den Rahmen des Shop-Modules. U.a. enthält diese Vorlage die Backtracking-Navigation ("Brotkrümel-Navigation") und Texte, die in allen Bereichen des Shops sichtbar sein sollen.

{cp:tags} für diese Vorlage	
{cp:shop:content}	Platzierung des Inhaltsbereiches innerhalb der Hauptvorlage des Shop-Modules.
{cp:shop:level}	Komplette Backtracking-Navigation für den CP::Shop.

5.6.120. Vorlagen "container/sections.html" und "container/sections_sub.html"

Diese Vorlagen liegen im Ordner "/modules/shop/templates/container/" und dienen der Anzeige der Kategorienstruktur als Navigation.

{cp:tags} für diese Vorlage	
{cp:shop:}*	Alle Angaben zur Kategorie aus der Datenbanktabelle *_shop_sections.
{cp:shop:numitems}	Anzahl der Artikel in dieser Kategorie.

5.6.121. Vorlagen "details.html", "list.html" und andere Artikeldarstellungen

Diese Vorlagen liegen im Ordner "/modules/shop/templates/" bzw. in den Unterordnern.

{cp:tags} für diese Vorlage	
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items.
{cp:shop:item:attributes}	Darstellung der Attribute der Artikelklasse, sofern vergeben.
{cp:shop:item:attributes:}*	Attribute der Artikelklasse, für "*" wird die Position des Attributs angegeben ("1" für 1. Attribut usw.).
{cp:shop:item:company_name}	Name des zugeordneten Herstellers.
{cp:shop:item:currency}	Kürzel der aktuell ausgewählten Währung.
{cp:shop:item:has_attributes}	Gibt an, ob Interaktive Attribute vorhanden sind oder nicht.
{cp:shop:item:loggedin}	Kunde eingeloggt ("1") oder nicht eingeloggt ("0").
{cp:shop:item:next_link}	Link zum vorherigen Artikel in dieser Kategorie.
{cp:shop:item:prev_link}	Link zum nächsten Artikel in dieser Kategorie.
{cp:shop:item:pricetrend}	Preisentwicklung aufwärts ("up") oder abwärts ("down").
{cp:shop:item:price_recommended}	Unverbindliche Preisempfehlung (UVP) des Artikels als Nettopreis.
{cp:shop:item:url}	URL zur Detailseite des Artikels.
{cp:shop:price:baseprice}	Grundpreis des Artikels, wenn ein Gewicht in Gramm angegeben wurde.
{cp:shop:price:discount_price}	Nettopreis des Artikels inkl. eventueller Kundenrabatte.
{cp:shop:price:discount_taxprice}	Bruttopreis des Artikels inkl. eventueller Kundenrabatte.
{cp:shop:price:is_discount}	Gibt an, ob für den Artikel ein Rabatt verfügbar ist (bisheriger Preis > aktueller Preis).
{cp:shop:price:oldprice}	Bisheriger Nettopreis des Artikels.
{cp:shop:price:oldtaxprice}	Bisheriger Bruttopreis des Artikels.
{cp:shop:price:price}	Nettopreis des Artikels.
{cp:shop:price:recommended_price}	Unverbindliche Preisempfehlung (UVP) des Artikels als Nettopreis.
{cp:shop:price:recommended_taxprice}	Unverbindliche Preisempfehlung (UVP) des Artikels als Bruttopreis.
{cp:shop:price:taxprice}	Bruttopreis des Artikels.
{cp:shop:price:taxrate}	MwSt.-Satz in Prozent (z.B. "19").

5.6.122. Vorlagen "print_html.html" und "print_text.html"

Diese Vorlagen liegen im Ordner "/modules/shop/templates/" und dienen der Darstellung der Rechnungen im HTML/Text-Format.

{cp:tags} für diese Vorlage	
{cp:shop:client:}*	Alle Angaben zum Kunden (Rechnungsadresse) aus der Datenbanktabelle *_shop_clients.
{cp:shop:client:hello}	Ansprache des Kunden (Rechnungsadresse), abhängig vom Geschlecht.
{cp:shop:item:}*	Alle Angaben zum Artikel aus der Datenbanktabelle *_shop_items.
{cp:shop:item:amount}	Anzahl der Einheiten des Artikels.
{cp:shop:item:price}	Nettopreis des Artikels.
{cp:shop:item:singleprice}	Nettopreis für eine Einheit des Artikels.
{cp:shop:item:tax}	Mehrwertsteueranteil für diesen Artikel.
{cp:shop:item:taxprice}	Bruttopreis des Artikels.
{cp:shop:options:}*	Allgemeine Einstellungen des CP::Shop (auch über "{cp:configuration:shop:*}" abrufbar).
{cp:shop:order:}*	Alle Angaben zur Bestellung aus der Datenbanktabelle *_shop_order.
{cp:shop:price:tax_*}	Mehrwertsteueranteil zum jeweiligen Steuersatz (z.B. "{cp:shop:price:tax_19}" für 19%).
{cp:shop:price:coupon}	Bruttorabatt für einen eventuell vorhandenen Gutschein.
{cp:shop:price:final}	Bruttogesamtpreis der gesamten Bestellung.
{cp:shop:price:options_price}	Nettopreis für die Kosten der gewählten Versand- und Zahlungsart.
{cp:shop:price:options_tax}	Mehrwertsteueranteil an den Kosten der gewählten Versand- und Zahlungsart.
{cp:shop:price:options_taxprice}	Bruttopreis für die Kosten der gewählten Versand- und Zahlungsart.
{cp:shop:price:price}	Nettozwischenpreis ohne Versand- und Zahlungsart, ohne eventuellen Gutschein.
{cp:shop:price:tax}	Mehrwertsteueranteil am Bruttozwischenpreis.
{cp:shop:price:taxprice}	Bruttozwischenpreis ohne Versand- und Zahlungsart, ohne eventuellen Gutschein.
{cp:shop:sendto:}*	Alle Angaben zum Kunden (Lieferadresse) aus der Datenbanktabelle *_shop_clients.
{cp:shop:sendto:hello}	Ansprache des Kunden (Lieferadresse), abhängig vom Geschlecht.

5.6.123. Vorlagen "print_label.html" und "print_packaging.html"

Diese Vorlagen liegen im Ordner "/modules/shop/templates/" und dienen der Ausgabe von Packlisten und Etiketten.

{cp:tags} für diese Vorlage	
{cp:shop:content}	Platzierung des Inhalts für die Druckvorlagen etc.

5.6.124. Vorlagen "section.html" und "section_empty.html"

Diese Vorlagen liegen im Ordner "/modules/shop/templates/" und dienen der Darstellung von Kategorien mit und ohne Artikel.

{cp:tags} für diese Vorlage	
{cp:shop:}*	Alle Angaben zur Kategorie aus der Datenbanktabelle *_shop_sections.
{cp:shop:section:}*	Alle Angaben zur Kategorie aus der Datenbanktabelle *_shop_sections.
{cp:shop:companies}	Auswahlformular für die in einer Kategorie verfügbaren Hersteller.
{cp:shop:companies_list}	Kommaseparierte Auflistung der ID's der Hersteller in einer Kategorie.
{cp:shop:company}	Aktueller Wert des Parameters "company".
{cp:shop:dir}	Aktueller Wert des Parameters "dir".
{cp:shop:items}	Anzahl der Artikel in dieser Kategorie.
{cp:shop:link}	Teil der URL zum Umblättern mit der Angabe der aktuellen Seite.
{cp:shop:numitems}	Anzahl der in dieser Kategorien insgesamt vorhandenen Artikel.
{cp:shop:numpages}	Anzahl der insgesamt verfügbaren Seiten mit Artikeln.
{cp:shop:order}	Aktueller Wert des Parameters "order".
{cp:shop:page}	Nummer der aktuell angezeigten Seite.
{cp:shop:pages}	Navigation zum Umblättern bei mehrseitigen Kategorien.
{cp:shop:per_page}	Aktueller Wert des Parameters "per_page".
{cp:shop:sectionin}	Kommaseparierte Liste der ID's der Unterkategorien.
{cp:shop:showitems}	Infos zu den aktuell angezeigten Artikeln ("von n1 bis n2...").
{cp:shop:show}	Ausgewählte Vorlage (standardmäßig leer bzw. "gallery").

5.6.125. Vorlagen im Ordner "info"

Diese Vorlagen liegen im Ordner "/modules/shop/templates/info/" und dienen der Anzeige von Informationen zu den AGB, Impressum, dem Widerrufsrecht uvm.

{cp:tags} für diese Vorlage	
{cp:shop:currency}	Kürzel der aktuell ausgewählten Währung.

5.6.126. Webservices: Allgemeines

Das Contentpapst-Framework, auf welchem CP::Shop aufgebaut ist, verfügt über eine professionelle Unterstützung für Webservices. Über Webservices können Informationen in festgelegten Formaten zwischen XML-fähigen Betriebssystemen/Programmen ausgetauscht und in beliebigen Programmiersprachen (PHP, ASP, VBasic, JSP usw.) weiterverwendet werden. Für Ihren mit CP::Shop gestalteten Online-Shop bedeutet dies konkret, dass beispielsweise angemeldete Partner auf ihren eigenen Webseiten neue Shops errichten können. Diese Shops können beliebig strukturiert und gestaltet werden, die notwendigen Daten werden live von CP::Shop ausgeliefert.

CP::Shop wertet alle von Ihnen eingetragenen Artikel und Kategorien aus und stellt diese strukturiert zum Abruf über HTTP (per URL über den Browser) zur Verfügung. Es wird sichergestellt, dass nur freigeschaltete und verfügbare Artikel angezeigt werden. Per Webservices sind so u.a. Daten zu Artikeln, Kategorien und Händlern abrufbar.

Über spezielle Parameter in der URL kann die Anzeige der Daten im XML-Format beeinflusst werden. Beispielsweise. können Artikel nach Titel, Preis, der EAN/UPC-Nummer, der Artikelnummer, Verkaufsrang und Bewertung geordnet werden. Durch ein vorangestelltes "+" oder "-" wird die Richtung der Ordnung beeinflusst. Zudem können über die Webservices spezielle RSS-/Produktdaten-Feeds für die Partner Ihres Partnerprogrammes bereitgestellt werden.

5.6.127. Webservices: Artikel

Der wichtigste Grund für den Einsatz von Webservices ist natürlich die Verwendung der aktuellen Artikeldaten auf anderen Webseiten. Daher ist dieser Aspekt hier besonders ausführlich beschrieben.

Die Abfrage der Artikel kann durch eine Reihe von Suchmöglichkeiten gezielt eingeschränkt werden. Angezeigt werden allerdings stets nur Artikel, die vorhanden (d.h. Lagerbestand mind. 1 bzw. unendlich) und öffentlich zugänglich sind. Z.B. können einzelne Begriffe im Titel gesucht werden oder es sollen nur Artikel eines bestimmten Herstellers angezeigt werden. Dies alles und noch viel mehr lässt sich mit den folgenden Parametern bewerkstelligen:

Auswahl des gewünschten Suchmodus	
searchnumber	Suche von Artikeln nach der internen Nummer (ID)
searchitem	Suche von Artikeln nach einer Artikelnummer
searchean	Suche nach einer EAN/UPC-Nummer
searchkeyword	Stichwort-Suche in Titel, Beschreibung und Schlüsselwörtern
searchsimilar	Suche nach ähnlichen Artikeln (mittels der internen Nummer)
searchtitle	Suche im Titel (Suchbegriff ist Teil des Titels)
searchmanufacturer	Name oder Nummer (ID) des Herstellers

searchsection	Suche von Artikeln in einer Kategorie (mittels der Kategorie-Nummer)
searchwishlist	Suche von Artikeln in einer Wunschliste (mittels der Nummer der Wunschliste)
rssfeed	Ausgabe der neuesten "n" Artikel im Shop (mit "limit" kann die Anzahl festgelegt werden)

Es muss immer genau einer dieser Parameter vorhanden sein, damit die Suche ordnungsgemäß durchgeführt werden kann. Neben diesem Parameter, der grundsätzlich den Suchmodus festlegt, können weitere Parameter angegeben werden:

Optionale Parameter für die Suche	
order	sortiert die zutreffenden Artikel (siehe unten)
page	gewünschte Seite festlegen (da max. 10 Artikel pro Seite)
strict	wenn "strict" gesetzt ist, dann muss die Suche exakt zutreffen

Sortiert werden kann nach "number" (interne Artikelnummer), "itemnr" (Artikelnummer im Shop), "ean_upc" (EAN/UPC-Nummer), "price" (Preis netto), "salesrank" (Verkaufsrang), "reviews" (Anzahl der Kundenmeinungen), "date" (Datum) und "title" (Titel). Eine Sortierung ist dabei immer in aufsteigender und absteigender Reihenfolge möglich und wird durch "+" bzw. "-" bestimmt (z.B. "+reviews" listet den am häufigsten bewerteten Artikel zuerst). Standardmäßig wird alphabetisch und aufsteigend sortiert.

In den Knoten "totalresults" und "totalpages" wird - wie bei anderen Abfragen - die Gesamtanzahl an Artikeln bzw. die Anzahl der Seiten angegeben. Diese Informationen können genutzt werden, um beispielsweise eine Navigation zum Aufruf der nachfolgenden Seiten zu realisieren.

5.6.128. Webservices: Artikel / Artikeldaten

Für die einzelnen Artikel werden im Knoten ("items") u.a. die folgenden Daten zurückgegeben. Je nach Artikel können diese allerdings unterschiedlich ausfallen (Kundenmeinungen vorhanden ja/nein usw.):

- *number* - Interne Nummer des Artikels (als Attribut)
- *itemnr* - Artikelnummer
- *ean_upc* - EAN/UPC-Nummer (nur wenn vorhanden)
- *titel* - Titel
- *short_description* - kurze Beschreibung
- *full_description* - ausführliche Beschreibung
- *availability* - Angabe zur Verfügbarkeit (Lieferstatus)
- *price* - Preis (inkl. MwSt.)
- *oldprice* - vorheriger Preis (inkl. MwSt., nur wenn vorhanden)
- *currency* - Währungskürzel (z.B. "EUR")
- *weight* - Gewicht (nur wenn vorhanden)
- *volume* - Größe (nur wenn vorhanden)
- *keywords* - Schlüsselwörter
- *image* - URL des primären Bildes des Artikels (nur wenn vorhanden)
- *manufacturer* - Knoten für Hersteller (mit Daten für "number" und "name")
- *salesrank* - Verkaufsrang (nur wenn bereits verkauft)
- *url* - URL zur Detailseite im Shop
- *similar* - Knoten mit ähnlichen Artikeln (jeweils Angabe von Nummer und Titel)
- *sections* - Knoten mit den Kategorien des Artikels (jeweils Angabe von Nummer und Titel)
- *reviews* - Knoten mit Kundenmeinungen zum Artikel (siehe Anmerkung unten)

Der Knoten "reviews" enthält die einzelnen Kundenmeinungen jeweils mit den Angaben von Titel, Beschreibung, Bewertung und Datum. Zusätzlich gibt es die Knoten "totalreviews" (Anzahl der Kundenmeinungen) und "averagerating" (durchschnittliche Bewertung). Allerdings sind hier maximal 3 Kundenmeinungen enthalten. Um mehr Kundenmeinungen abzurufen, kann ein separater Abruf durchgeführt werden (siehe "Webservices / Artikel / Kundenmeinungen").

5.6.129. Webservices: Artikel / Beispiele

Hier finden Sie einige Beispiele zur Zusammenstellung der Parameter für die Suchanfrage mittels Webservices:

```
*/webservices.php?file=shop&searchsection=43&page=2
// 2. Seite der Artikel in der Kategorie "43" (angezeigt werden maximal 10 auf einmal)
```

```
*/webservices.php?file=shop&searchtitle=tennis&hide=full_description|short_description
// Artikel mit Tennis im Titel (diese werden ohne die kurze und lange Beschreibung geliefert)
```

```
*/webservices.php?file=shop&searchmanufacturer=microsoft&order=+price&strict=yes
// es werden alle Artikel von "Microsoft" nach dem Preis geordnet geliefert
// (der Hersteller der Artikel muss dabei unbedingt "Microsoft" sein und nicht z.B. "Microsoft Germany")
```

Durch die Variierung der einzelnen Parameter ("searchitem" usw., "order", "page", "strict" und "keyword") haben Sie somit zahlreiche Möglichkeiten, gezielt die gewünschten Artikel von CP::Shop geliefert zu bekommen.

5.6.130. Webservices: Artikel / Einschränkung der Artikeldaten

Damit nicht bei jedem Abruf alle Artikeldaten abgerufen werden müssen, können Sie die Auswahl der zurückgelieferten Daten mittels des Parameters "hide" eingrenzen. In diesem sind die Knoten aufgelistet, deren Inhalt Sie nicht benötigen. Möglich sind:

```
"short_description", "full_description", "ean_upc", "availability", "weight", "volume", "keywords", "image", "similar",
"reviews", "sections", "manufacturer" und "salesrank"
```

Der Parameter "hide" wird einfach in die URL an einer beliebigen Stelle eingefügt. Diese Einschränkung auf bestimmte Knoten kann die Abfrage teilweise erheblich beschleunigen (z.B. wenn nur eine Übersicht mit URL und Titel erstellt werden soll):

```
*/webservices.php?file=shop&searchtitle=contentpapst&hide=full_description|short_description
// sucht nach Contentpapst und liefert alles außer den Beschreibungen zurück
```

5.6.131. Webservices: Artikel / Kundenmeinungen

Zusätzlich zu den direkt in den Artikelabfragen integrierten Kundenmeinungen (max 3 pro Artikel), können diese auch separat abgerufen werden. Hier benötigen Sie die Artikel-Nummer, welche Sie an den Parameter "reviews" anhängen:

```
*/webservices.php?file=shop&reviews=388291 // Kundenmeinungen für den Artikel "388291"
```

Mit einer derart aufgebauten Abfrage werden Ihnen maximal 10 Kundenmeinungen zu einem Artikel pro Seite angezeigt. Die Struktur der Kundenmeinungen entspricht dabei der aus den Artikelabfragen bekannten Struktur (d.h. mit "totalreviews" und "averagerating"). Zusätzlich wird Ihnen über "totalpages" die Seitenanzahl angezeigt.

Um weitere Seiten anzuzeigen oder die Ausgabe anderweitig zu beeinflussen, nutzen Sie die Parameter "page" (aktuelle Seite) und "order" (Sortieren nach "+title", "-title", "+date", "-date", "+rating", "-rating"). Standardmäßig wird nach dem Datum sortiert mit der neuesten Kundenmeinung am Anfang.

5.6.132. Webservices: Darstellung über XSLT

Alle per Webservices erreichbaren Inhalte können frei per XSLT formatiert werden. Hierzu muss CP::Shop über einen weiteren Parameter "xsl" lediglich der Pfad zur benötigten XSL-Datei mitgeteilt werden. Die XML-Ausgabe wird dann entsprechend den in der XSL-Datei getroffenen Regeln formatiert. CP::Shop setzt den Content-Type für die Darstellung automatisch auf "text/html", alternativ kann an die URL auch "&ct=text/html" angehängt werden.

Ein Beispiel für den Aufruf mit Pfad zu einer XSL-Datei:

```
webservices.php?file=shop&searchtitle=sport&ct=text/html&xsl=http://www.domain.de/sport.xsl
// Suche nach "Sport" mit der dazu passenden XSL-Datei "sport.xsl"
```

Diese Datei könnte z.B. folgendermaßen aufgebaut sein für eine einfache Auflistung der gefundenen Suchergebnisse mit der Anzahl der insgesamt gefundenen Artikel und dem Link zur Detailseite:

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/TR/xhtml1/strict">

<xsl:output method="xhtml" version="1.0" encoding="ISO-8859-1" indent="no" omit-xml-declaration="yes" doctype-
public="-//W3C/DTD XHTML 1.0 Transitional//EN" doctype-system="
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" standalone="yes" media-type="text/html" />

<xsl:template match="/">

<html>

<head>
</head>

<body>

<u>Meine erste XSL-Datei (insgesamt <xsl:value-of select="shop/totalresults" /> Ergebnisse)</u><br /><br />

<xsl:for-each select="shop/items/item">

<strong><xsl:value-of select="title"/></strong><br /><br />
<xsl:value-of select="short_description"/> <br /><br />

<xsl:text disable-output-escaping="yes"><</xsl:text>a href="<xsl:value-of select="url"/>"<xsl:text disable-output-
escaping="yes">></xsl:text>mehr Informationen ueber diesen Artikel...<xsl:text disable-output-
escaping="yes"><</xsl:text>/a<xsl:text disable-output-escaping="yes">></xsl:text>

<hr/>

</xsl:for-each>

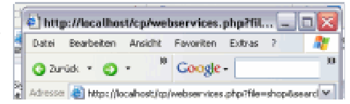
</body>

</html>

</xsl:template>

</xsl:stylesheet>
```

Das Ergebnis dieser Umwandlung mittels XSLT sehen Sie auf der rechten Seite. Da es an dieser Stelle nicht möglich ist, auch nur einen Überblick über die Leistungsfähigkeit von XSL/XSLT zu geben, sei auf die hervorragende Seite des W3C verwiesen. Sie finden unter http://www.w3.org/Style/XSL/ die



offizielle Spezifikation zu XSLT und einige nützliche Hinweise zur Verwendung.

XSLT ermöglicht Ihnen u.a.:

- die freie Darstellung von XML-Daten in HTML/XHTML
- Umwandlung in andere Dateiformate (z.B. für den Datenaustausch)

Bitte beachten Sie, dass PHP für den Einsatz von XSLT über die XSLT-Extension verfügen muss!

5.6.133. Webservices: Erweiterte Suche

Die erweiterte Suche ("prosearch") erlaubt eine noch weitreichendere Beeinflussung der Suchergebnisse. Über den Parameter "prosearch" können Sie an CP::Shop eine komplexe Suchabfrage übermitteln, welche u.a. boolsche Verknüpfungen mit einschließen ("and", "or" und "not") kann.

Gesucht wird über eine Verknüpfung von Feldname und gewünschtem Inhalt, z.B.:

```
keywords%3Asport+and+keywords%3Atennis+and+price-max%3A20.00
// alle Artikel für max. 20,00 EUR, die die Schlüsselwörter "Sport" und "Tennis" enthalten
description%3Aschumacher+or+description%3Amontoya+or+title%3Aformel1
// Beschreibung muss "schumacher" oder "montoya" enthalten oder der Titel muss "Formel1" enthalten
```

Mögliche Felder sind hier "description", "title-starts", "title-ends", "title", "title-exact", "ean_upc", "itemnr", "keywords", "price-max" und "price-min" die über die boolschen Verknüpfungen "and", "or" und "not" miteinander kombiniert werden können. Bitte beachten Sie, dass der Inhalt des Parameters "prosearch" unbedingt URL-kodiert sein muss, d.h. Leerzeichen sind durch "+" zu ersetzen , ":" durch "%3A" usw.

Hier zwei kleine Beispiele für die Benutzung der erweiterten Suche:

```
webservices.php?file=shop&prosearch=title-starts%3Afernsehen+not+keywords%3Acnn
// zeigt alle Artikel mit "fernsehen" am Anfang des Titels, "cnn" darf nicht in den Schlüsselwörtern enthalten sein
webservices.php?file=shop&prosearch=price-max%3A200.00&order=-price
// zeigt alle Artikel die günstiger als 200,00 EUR sind
```

5.6.134. Webservices: Hersteller-Informationen

Über jeden Hersteller kann anhand der Nummer oder des Namens eine Information eingeholt werden, welche die folgenden Daten beinhaltet (im Knoten "manufacturer"):

- *title* - Bezeichnung / Firmierung des Unternehmens
- *description* - Beschreibung
- *website* - die Webseite des Unternehmens
- *street/plz/city/country* - Angaben zur Adresse des Unternehmens

Darüberhinaus werden - sofern für den Hersteller vorhanden - max. 10 Artikel mit Nummer und Titel angezeigt (im Knoten "items"). Diese müssen freigeschalten und vorrätig sein. Der Knoten "numitems" liefert zudem die gesamte Anzahl der Artikel des Herstellers. Wird zusätzlich der Parameter "variations" mit dem Wert "1" angehängt, so werden in die Artikel auch Varianten mit einbezogen.

```
*/webservices.php?file=shop&manufacturerinfo=29691 // alle Artikel des Herstellers mit der Nummer "29691"
*/webservices.php?file=shop&manufacturerinfo=macromedia // Artikel des Herstellers "Macromedia"
```

5.6.135. Webservices: Kategorien

Die Struktur der Kategorien wird über die Webservices ebenfalls abgebildet. Wie bei den Herstellern erfolgt auch hier wieder der Aufruf anhand der Nummer oder des Namens, diesmal jedoch von der darüberliegenden Kategorie (Oberkategorie). Sofern diese Kategorie Unterkategorien besitzt, werden im Knoten "sections" die folgenden Informationen ausgegeben:

- *number* - Nummer der Kategorie (als Attribut des Knotens)
- *title/description* - Titel und Beschreibung der Kategorie
- *url* - direkter Link zur Kategorieansicht im Shop

Der Knoten "totalresults" liefert zusätzlich die gesamte Anzahl der Unterkategorien der aktuellen Kategorie. Da max. 10 Unterkategorien gleichzeitig ausgegeben werden, enthält der Knoten "totalpages" eine Angabe über die Seitenanzahl.

Über den optionalen Parameter "page" kann schließlich die gewünschte Seite festgelegt werden, order" sorgt für die richtige Sortierung der Ergebnisse (z.B. "+title", "-title", "+position", "-position", "+number" und "-number"). Standardmäßig wird die erste Seite angezeigt mit den alphabetisch geordneten Kategorien.

```
*/webservices.php?file=shop&sectioninfo=1&order=+title // Abfrage für die Kategorie "1"
*/webservices.php?file=shop&sectioninfo=213&order=+position&page=4 // Seite 4 der Unterkategorien von Kategorie "213"
*/webservices.php?file=shop&sectioninfo=all&order=+number // alle Kategorien, geordnet nach Nummern
```

5.6.136. Webservices: Suchwort-Filter

Die Auswahl der Artikel kann zusätzlich zu "searchsection", "searchtitle" usw. durch einen Suchwort-Filter verfeinert

werden. Hierzu wird der Parameter "keyword" mit dem gewünschten Suchwort an die URL angehängt. Der Suchwort-Filter durchsucht dann Titel, Beschreibung und Schlüsselwörter. Allgemein kann der Suchwort-Filter bei allen Abfragen genutzt werden, nicht jedoch mit den Parametern "prosearch", "searchkeyword", "searchwishlist" und "searchsimilar" sowie den Informationen zu Herstellern ("manufacturerinfo") und Kategorien ("sectioninfo").

5.6.137. Webservices: Warenkorb

Eine Besonderheit der Webservices-Schnittstelle von CP::Shop ist die Möglichkeit, den Warenkorb mittels Webservices verwalten zu können, ohne dass der Kunde bemerkt, dass dieser Vorgang eigentlich synchron bei einem anderen Shop abläuft. Um eine Bestellung mittels Webservices durchzuführen, sind folgende Schritte nacheinander zu befolgen:

- Neuen Warenkorb anlegen
- Warenkorb mit Artikeln füllen
- Artikelanzahl anpassen (optional)
- Artikel entfernen (optional)
- Warenkorbinhalt anzeigen (optional)
- Warenkorb löschen (optional)
- Warenkorb absenden

Neuen Warenkorb anlegen

Bevor Artikel zum Warenkorb hinzugefügt werden können, muss zuerst ein neuer Warenkorb über eine spezielle URL angelegt werden. U.a. wird in dieser URL der Parameter "cart" genutzt mit dem Wert "create". Als Ergebnis liefert CP::Shop einen "Userkey" zurück, der nachfolgend diesen Warenkorb repräsentiert.

```
*/webservices.php?file=shop&cart=create // legt einen Warenkorb an
```

Der Userkey sollte im Partner-Shop hinterlegt werden. Dies kann bspw. per Session, Cookie oder als Parameter über die URL geschehen. Wenn der Userkey nicht mehr vorhanden ist, dann kann auch keine Verbindung mehr zwischen dem aktuellen Nutzer und dessen Warenkorb hergestellt werden.

Warenkorb mit Artikeln füllen

Nachdem der Warenkorb angelegt und der Userkey gespeichert wurde, kann im nächsten Schritt mit dem Hinzufügen der neuen Artikel begonnen werden. Der Parameter "cart" in der URL besitzt nun den Wert "add". Schrittweise werden dem CP::Shop jetzt die einzelnen Artikel übermittelt, zusammen mit der gewünschten Anzahl und dem Userkey. Für jeden Artikel muss die shop-interne Nummer (aus dem Feld "number") bzw. die normale Artikelnummer (aus dem Feld "itemnr") übermittelt werden.

```
*/webservices.php?file=shop&cart=add&article.[ARTIKEL-NUMMER]=[ANZAHL]&userkey=[USERKEY]
```

```
*/webservices.php?file=shop&cart=add&article.CP_015=2&userkey=3jlkjf8JLKAAi95
// Artikel "CP_015" zu einem vorhandenen Warenkorb hinzufügen
```

Sie können gleichzeitig beliebig viele unterschiedliche Artikel zum Warenkorb hinzufügen, indem der Abschnitt "&article.[ARTIKEL-NUMMER]=[ANZAHL]" in der URL mit den entsprechenden Angaben wiederholt wird.

Artikelanzahl anpassen

Im Warenkorb vorhandene Artikel können jederzeit in der Anzahl geändert werden. Hierzu muss in der URL der Parameter "cart" auf "edit" gesetzt werden. Es folgt die Aufzählung der Artikel, zusammen mit der gewünschten neuen Anzahl:

```
*/webservices.php?file=shop&cart=edit&article.[ARTIKEL-NUMMER]=[ANZAHL]&userkey=[USERKEY]
```

Die neue Anzahl kann eine beliebige Zahl > 0 sein. Um einen Artikel aus dem Warenkorb komplett zu entfernen, nutzen Sie den folgenden Schritt. Diese Trennung dient hauptsächlich der besseren Übersicht.

Artikel entfernen

Ein Artikel kann auch komplett aus dem Warenkorb entfernt werden. Der Parameter "cart" lautet nun "delete". Es folgt die Aufzählung der Artikel, zusammen mit der gewünschten neuen Anzahl (wie bei "add" und "edit"):

```
*/webservices.php?file=shop&cart=delete&article.[ARTIKEL-NUMMER]=[ANZAHL]&userkey=[USERKEY]
```

Warenkorbinhalt anzeigen

Zur Simulation eines Warenkorbes gehört natürlich auch die Anzeige des aktuellen Warenkorbinhaltes mit den einzelnen Artikeln und deren Menge. Diese Informationen können z.B. mit der Abfrage von Artikelinformationen verknüpft werden, um Preise und Bilder im Warenkorb anzeigen zu können. Der Parameter "cart" lautet nun "get":

```
*/webservices.php?file=shop&cart=get&userkey=[USERKEY]
```

Warenkorb löschen

Um sich beim Löschen des Warenkorbinhaltes das schrittweise Löschen der einzelnen Artikel zu ersparen, gibt es auch für diese Aufgabe eine spezielle URL. Der Parameter "cart" wird nun auf "clear" gestellt:

```
*/webservices.php?file=shop&cart=clear&userkey=[USERKEY]
```

Warenkorb absenden

Sind alle Artikel im Warenkorb, können Sie den Benutzer mittels Checkout-Button zum eigentlichen Shop springen lassen. Dort kann er durch Auswahl der gewünschten Zahlungsart, Eintragen seiner Adresse usw. dann die Bestellung abschließen. Der Parameter "cart" muss hierzu auf "checkout" gestellt werden:

```
*/webservices.php?file=shop&cart=checkout&userkey=[USERKEY]
```

5.6.138. Webservices: Wunschlisten

Für die Abfrage der Wunschlisten gilt die Anleitung unter "Webservices / Artikel". Eine Besonderheit bei den Wunschlisten sei jedoch noch erwähnt: Die einzelnen Artikel einer Wunschliste werden stets nach dem Datum sortiert, an dem diese der Wunschliste hinzugefügt werden. Der Artikel mit dem neuesten Datum steht dabei immer am Anfang. Zusätzlich weist jeder Artikel ein Feld "date_added" auf, in dem das passende Datum zu finden ist.

5.6.139. Übersicht der vom Shop genutzten Vorlagen

Bevor Sie Ihren Shop offiziell eröffnen, sollten Sie zuerst alle Vorlagen im Ordner "/modules/shop/templates/" anpassen. Dies sind u.a. Vorlagen für die Gestaltung von Formularen, Rechnungen, Etiketten und Emails. In den Standardvorlagen sehen Sie bereits, welche {cp:tags} verwendet werden können und wie diese einzusetzen sind. Wenn Ihnen der Verwendungszweck eines {cp:tags} unklar sein sollte, dann sollten Sie diesen Teil des Shops einfach aus der Benutzersicht live testen. So sehen Sie, wie sich dies später auf Ihre Besucher und Kunden auswirken wird. Für einzelne Vorlagen und Container gibt es in dieser Anleitung eine ausführlichere Erläuterung der jeweils verfügbaren {cp:tags}.

Im Folgenden sehen Sie eine Auflistung der standardmäßig mitgelieferten Vorlagen im Ordner "/modules/shop/templates/":

Vorlagen direkt im Hauptordner "templates" (für verschiedene Grundfunktionen etc.)	
"details.html"	Standardvorlage für die Detail-Darstellung eines Artikels im Shop.
"export_html.html"	Vorlage für den HTML-Export von Artikeln.
"list.html"	Standardvorlage für die Listen-Darstellung der Artikels im Shop.
"list_cart.html"	Standardvorlage für die Listen-Darstellung der Artikel im Warenkorb.
"list_comparison.html"	Standardvorlage für die Listen-Darstellung im Artikelvergleich.
"list_gallery.html"	Standardvorlage für die Galerie-Darstellung der Artikel im Shop.
"login.html"	Vorlage für den Login in den Kunden-Bereich.
"logout.html"	Meldung, welche erscheint, sobald sich ein Kunde ausgeloggt hat.
"maintenance.html"	Meldung, welche erscheint, wenn sich der Shop im Wartungsmodus befindet.
"message.html"	Standardvorlage für die Ausgabe einer Meldung an den Besucher / Kunden.
"print_details.html"	Vorlage für die Druckversion einer einzelnen Artikels.
"print_html.html"	Druck einer Rechnung im HTML-Format (vom Administrationsbereich aus).
"print_label.html"	Druck der Etiketten für einzelne / mehrere Bestellungen (vom Administrationsbereich aus).
"print_packaging.html"	Druck einer Packliste (vom Administrationsbereich aus).
"print_text.html"	Druck einer Rechnung im Text-Format (vom Administrationsbereich aus).
"recommend.html"	Formular für die Weiterempfehlung von Artikeln.
"recommend_success.html"	Anzeige einer Meldung nach dem Versand der Weiterempfehlung.
"recovery.html"	Formular für die Anforderung eines vergessenen Passwortes.
"recovery_send.html"	Anzeige einer Meldung nach dem Versand des obigen Formulares.
"register.html"	Formular für die Anmeldung eines Kunden unabhängig von einer Bestellung.
"related.html"	Vorlage für die Anzeige von ähnlichen Artikeln ("Kunden, die x gekauft haben...").
"search.html"	Vorlage für die Anzeige der Suchfunktion.
"section.html"	Standardvorlage für die Darstellung einer Kategorie im Shop.
"section_empty.html"	Standardvorlage für die Darstellung einer leeren Kategorie im Shop.
"site.html"	Hauptvorlage. Über diese wird das Layout des Shops bestimmt (wichtig).
"special.html"	Standardvorlage für Artikel, die als Sonderangebot markiert wurden.

Vorlagen im Ordner "articles" (für die Darstellung der Artikel-Detailseiten)
Welche Vorlagen in diesem Ordner enthalten sind, ist abhängig davon, welche Vorlagen für die Darstellung der Shop-Artikel benötigt werden. Es kann hier eine beliebige Anzahl an Vorlagen angelegt werden, die dann in den Formularen für die Artikel zur Auswahl bereitgestellt werden (zusätzlich zur Standardvorlage "/modules/shop/templates/details.html").

Vorlagen im Ordner "cart" (für den Warenkorb / die Bestellung)	
"cart/cart_0.html"	Startseite des Warenkorbs mit einer Auflistung der momentan enthaltenen Artikel.
"cart/cart_1.html"	Zweite Seite im Bestellprozess. Hier wird ein Login durchgeführt oder ein neues Kunden-Konto angelegt.
"cart/cart_2.html"	Dritte Seite im Bestellprozess. Es wird die gewünschte Zahlungs- und Versandart ausgewählt.
"cart/cart_3.html"	Abschluss des Bestellprozesses mit einer Übersicht der gemachten Angaben.
"cart/cart_empty.html"	Gibt eine Meldung aus, sobald der Warenkorb durch den Kunden geleert wurde.
"cart/cart_express.html"	Möglichkeit zur Express-Bestellung für einen bereits im Shop eingeloggten Kunden.
"cart/cart_sendto.html"	Optionales Formular für die Eingabe einer zusätzlichen Versandanschrift.
"cart/checkout.html"	Optionale Vorlage auf der Basis von "/modules/shop/templates/message.html" für die Bestellbestätigungsseite.

Vorlagen im Ordner "clients" (für den Kunden-Bereich)

"clients/account.html"	Startseite des Kundenbereiches mit der Übersicht der bisherigen Bestellungen.
"clients/edit_address.html"	Verwaltung der eigenen Lieferadressen durch den Kunden.
"clients/edit_order.html"	Bearbeitung einer Bestellung / Bestellung stornieren usw.
"clients/edit_password.html"	Bearbeitung der eigenen Passwörter durch die Kunden.
"clients/edit_payment.html"	Bearbeitung der Zahlungseinstellungen durch die Kunden.
"clients/edit_wishlist.html"	Bearbeitung des eigenen Wunschzettels durch die Kunden.
"clients/options.html"	Formular für den Kunden zum Ändern der eigenen Angaben (Anschrift etc.).

Vorlagen im Ordner "container" (für die Formatierung der Container)	
"container/addsmart.html"	Formular zum schnellen Hinzufügen von Artikeln zum Warenkorb mittels Artikel-Nummer.
"container/companies.html"	Standardvorlage für die Darstellung von Herstellerlisten und -details.
"container/crossselling.html"	Vorlage für die Darstellung der Artikelverknüpfungen.
"container/delivery.html"	Zeigt im Shop die unterschiedlichen Möglichkeiten für den Lieferstatus der Artikel an.
"container/loggedin.html"	Vorlage für die Login-Box. Wird angezeigt, wenn der Kunde eingeloggt ist.
"container/login.html"	Vorlage für die Login-Box. Wird angezeigt, wenn der Kunde nicht eingeloggt ist.
"container/media.html"	Vorlage für die Darstellung der Mediendateien (Auflistung).
"container/media_gallery.html"	Vorlage für die Darstellung der Mediendateien (Galerie).
"container/owner_rating.html"	Vorlage für die Darstellung der Bewertung durch den Shop-Betreiber.
"container/payment.html"	Vorlage für die detaillierte Auflistung der Zahlungsarten.
"container/prices.html"	Vorlage für die Darstellung der Preisstaffel eines Artikels.
"container/pricetrend.html"	Vorlage für den Aufbau des Preistrends für die Artikel.
"container/review_box.html"	Anzeige eines Formulares für die Rezension von Artikeln.
"container/reviews.html"	Ausgabe der Rezensionen zu einem Artikel.
"container/search.html"	Formular für den Zugriff auf die Suchfunktion des Shops.
"container/sections.html"	Formatierung der Anzeige der Kategorien / Kategoriestruktur.
"container/sections_sub.html"	Formatierung der Anzeige der Unterkategorien / Kategoriestruktur.
"container/shipment.html"	Vorlage für die detaillierte Auflistung der Versandarten.
"container/variants.html"	Standardvorlage für die Darstellung der Artikelvarianten.

Vorlagen im Ordner "emails" (für den Versand von Emails)	
"emails/de.availability.tpl"	Email an den Shopbetreiber bei Artikeln mit geringem Warenbestand.
"emails/de.cancellation.tpl"	Email an den Shopbetreiber, bei Stornierungen von Artikeln / Bestellungen.
"emails/de.companies.tpl"	Email an einen Hersteller, wenn ein Artikel ausverkauft ist.
"emails/de.order.tpl"	Email an einen Kunden mit der Bestätigung der Bestellung (wichtig).
"emails/de.partner_buy_mail.tpl"	Email an einen Partner mit Informationen über eine neue (geworbene) Bestellung.
"emails/de.partner_free_mail.tpl"	Email an einen Partner bei Freischaltung mit Informationen über das Partnerprogramm.
"emails/de.partner_message.tpl"	Email an den Shop-Betreiber durch den Partner.
"emails/de.partner_recovery.tpl"	Email an einen Partner mit einem neu generierten Passwort.
"emails/de.partner_register.tpl"	Email an den Shopbetreiber bei der Anmeldung eines neuen Partners.
"emails/de.priceinfo.tpl"	Email an den Shop-Betreiber mit einem Wunschpreis.
"emails/de.recommend.tpl"	Vorlage für eine Email, welche bei der Weiterempfehlung eines Artikels verschickt wird.
"emails/de.recovery.tpl"	Email an einen Kunden mit einem neu generierten Passwort.
"emails/de.register.tpl"	Email an einen Kunden mit Informationen zu seinem Kunden-Konto nach der Anmeldung / Bestellung.
"emails/de.register_admin.tpl"	Email an den Administrator mit Informationen zur Anmeldung eines neuen Kunden.
"emails/de.return.tpl"	Email an den Shop-Betreiber mit einer RMA-/Rückgabe-Anforderung eines Kunden.
"emails/de.reviews.tpl"	Email an den Shopbetreiber mit der Benachrichtigung über neue Kundenmeinungen.
"emails/de.shipment.tpl"	Email an einen Kunden mit der Versandbestätigung zu seiner Bestellung.
	Weitere Vorlagen können je nach genutzten Zahlungsarten etc. vorhanden sein, bspw. als Bestätigung einer Zahlung über einen externen Zahlungsdienstleister (z.B. Paypal, mPAY24 oder T-Pay).

Vorlagen im Ordner "info" (für erweiterte Informationen)	
"info/aboutus.html"	Wir über uns
"info/agb.html"	Allgemeine Geschäftsbedingungen (AGB)
"info/companyinfo.html"	Informationen zu einem Hersteller
"info/companylist.html"	Auflistung der im Shop vertretenen Hersteller
"info/comparison.html"	Vergleich verschiedener Artikel
"info/contact.html"	Kontaktfomular
"info/feeds.html"	RSS-Feeds des Shops
"info/help.html"	Hilfe für die Besucher / Kunden
"info/imprint.html"	Impressum
"info/information.html"	Weitere Informationen
"info/new-products.html"	Neue Artikel im Shop
"info/philosophy.html"	Unsere Philosphie
"info/priceinfo.html"	Formular für die Anforderung eines Wunschpreises
"info/pricetrend.html"	Preistrends
"info/privacy.html"	Datenschutzerklärung

"info/return.html"	RMA-/Rückgabe-Formular
"info/rights.html"	Widerrufsrecht
"info/shipment.html"	Detaillierte Anzeige der Versandarten/Zahlungsarten.
"info/sitemap.html"	Sitemap
"info/special-products.html"	Sonderangebote
"info/trustedshops.html"	Informationsseite für Shops mit Trusted Shops Gütesiegel.
"info/wishlist.html"	Wunschzettel der Kunden
	Weitere Vorlagen können beliebig hinzugefügt werden. Lesen Sie dazu auch "Erweiterte Informationen im Shop einbinden".

Vorlagen im Ordner "partner" (für den Partner-Bereich)	
"partner/account.html"	Übersichtsseite nach dem Login.
"partner/ads.html"	Vorlage für die Darstellung der Werbemittel für die Partner.
"partner/edit_password.html"	Formular für die Bearbeitung des Passwortes.
"partner/login.html"	Vorlage für den Login der Partner.
"partner/logout.html"	Vorlage für den Logout der Partner.
"partner/message.html"	Vorlage für den Versand einer Nachricht an den Shop-Betreiber.
"partner/options.html"	Formular für die Bearbeitung der allgemeinen Kontakt-Daten.
"partner/order.html"	Vorlage für die Anzeige der bisherigen Provisionszahlungen.
"partner/partner_1.html"	Vorlage mit verschiedenen Fragen an den Partner vor einer Anmeldung.
"partner/partner_2.html"	Anmeldeformular für das Partnerprogramm.
"partner/partner_no.html"	Meldung, welche ausgegeben wird, wenn der Partner die gewünschten Kriterien nicht erfüllt.
"partner/partner_success.html"	Meldung, welche ausgegeben wird, wenn der Partner die Anmeldung erfolgreich abgeschlossen hat.
"partner/payment.html"	Formular für die Bearbeitung der Konto-Daten.
"partner/recovery.html"	Formular für die Anforderung eines vergessenen Passwortes.
"partner/recovery_send.html"	Vorlage nach der Anforderung eines vergessenen Passwortes.
"partner/statistic.html"	Vorlage für die Anzeige der Statistiken über Klicks / Verkäufe.

Vorlagen im Ordner "payment" (für die externen Zahlungsanbieter)
Welche Vorlagen in diesem Ordner enthalten sind, ist abhängig davon, welche Zahlungsschnittstellen mitgeliefert wurden. Je nach externem Zahlungsanbieter kann hier eine variable Anzahl an Vorlagen vorhanden sein.

Vorlagen im Ordner "sections" (für die Darstellung der Artikel-Listen)
Welche Vorlagen in diesem Ordner enthalten sind, ist abhängig davon, welche Vorlagen für die Darstellung der Artikel-Listen benötigt werden. Es kann hier eine beliebige Anzahl an Vorlagen angelegt werden, die dann in den Formularen für die Kategorien bereitgestellt werden (zusätzlich zur Standardvorlage "/modules/shop/templates/list.html").

Besonders wichtig ist eine ordnungsgemäße Gestaltung der Rechnungen. Die Standardvorlagen entsprechen soweit allgemein möglich den rechtlichen Vorschriften. Eine rechtlich verbindliche Beratung kann aber im Zweifelsfall hilfreich sein.

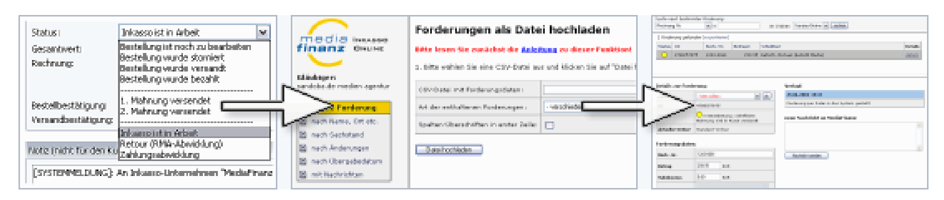
5.6.140. Zusatzmodul: Inkasso über mediafinanz AG



Die **mediafinanz AG** mit Sitz in Osnabrück ist für mittlerweile über 2.000 Mandanten der Partner für alle Fragen rund um das Thema Forderungsmanagement und Optimierung von Mahnabläufen. Die Mehrzahl der Kunden kommt aus dem Bereich Business to Consumer. Große Forderungsmengen sind hier genauso üblich, wie die Notwendigkeit, Zahlungsausfälle so gering wie möglich zu halten, um die Liquiditätslage des Unternehmens nicht zu verschlechtern.

Säumige Kunden / Bestellungen können über die Exportfunktionalität von CP::Shop direkt exportiert und ohne manuelle Nacharbeit bei MediaFinanz importiert werden. Sie sparen damit gerade bei einer größeren Anzahl an Schuldnern wesentliche Zeit, die Sie beispielsweise in Kundensupport reinvestieren können.

Um MediaFinanz als Inkasso-Unternehmen zu nutzen, ist zuerst eine Anmeldung unter http://www.mediafinanz.net/ notwendig. Anschließend müssen säumige Bestellungen den Status "Inkasso ist in Arbeit" zugewiesen bekommen. Unter "Bestellung bearbeiten" müssen Sie dann noch als alternatives Format "Inkasso MediaFinanz" wählen und Sie erhalten die notwendige Export-Datei mit allen Angaben als Download.



Im Gläubiger-Konto von MediaFinanz gehen Sie über "neue Forderungen als Datei hochladen" und wählen die gespeicherte Datei aus. Nun klicken Sie auf "Datei hochladen". Beim Import der ersten Forderungen müssen noch die Felder den einzelnen Spalten zugeordnet werden. Geben Sie dabei folgende Reihenfolge an:

Vorname | Nachname | Geschlecht | Firma | Strasse | PLZ | Ort | Land | Telefon 1 | Telefon 2 | Fax | Email | Rechnungsnr | Forderungsbetrag | Forderungsgrund | Leistungsdatum | Mahndatum | Forderungstyp

Bei jedem weiteren Import ist diese Zuordnung bereits voreingestellt, so dass die Forderungen direkt mit "Zuordnung stimmt >> weiter" importiert werden können. Mit dem Import beginnt MediaFinanz nun mit dem mehrstufigen Mahnsystem.

5.7. CP::Weblog

*CP::Weblog ist ein **kostenpflichtiges Zusatzmodul** für **Contentpapst**. Diese Informationen benötigen Sie daher nur, wenn Sie den CP::Weblog lizensiert haben und auf Ihrer Webseite einsetzen wollen. Weitere Informationen zu CP::Weblog erhalten Sie unter http://www.sandoba.de/produkte/blog-cpweblog/.*

5.7.1. Installation von CP::Weblog

CP::Weblog wird zusätzlich zu einer bestehenden Contentpapst-Installation als CP::Forms-Modul installiert. Das gelieferte ZIP-Archiv wird dafür zuerst extrahiert (inkl. der dort enthaltenen Ordner). Anschließend wird folgendermaßen vorgegangen:

1. Das Script "weblog_install.php" in das Hauptverzeichnis von Contentpapst kopieren.
2. Dieses Script über den Browser ausführen. CP::Weblog wird dabei im Hintergrund angelegt.
3. Nachdem das Script beendet ist, dieses vom Server löschen.
4. Anschließend die Ordner aus dem Archiv in den Ordner "modules" hochladen.
5. Innerhalb dieser hochgeladenen Ordner gibt es jeweils die Unterordner "files" und "templates". Für diese Ordner muss per FTP noch der CHMOD auf 775 oder 777 gesetzt werden, damit z.B. Bilder zu einer Blog-Eintrag hochgeladen werden können.

Nun kann das neue Modul über den Administrationsbereich unter "Module > CP::Forms" aufgerufen werden. Weblog-Einträge lassen sich nun bereits verfassen. Zuerst sollten Sie jedoch noch Kategorien anlegen und die Einstellungen von CP::Weblog an Ihre Bedürfnisse anpassen (u.a. ist hier die ID des Artikels einzutragen, in welchem der Weblog angezeigt wird).

5.7.2. Benutzung von CP::Weblog im Administrationsbereich

Da es bei CP::Weblog um ein CP::Forms-Modul handelt, gibt es im Administrationsbereich auch für dieses Modul den bekannten Komfort bei der Erstellung / Bearbeitung von Einträgen usw. D.h. Standardfunktionen für die Anpassung der Module, Im- und Export usw. stehen auch hier zur Verfügung (weitere Informationen sind dazu daher im Abschnitt "CP::Forms" zu finden).

Insgesamt besteht das Weblog aus den 5 Teilmodulen "Einträge", "BlogRoll", "Einstellungen", "Kategorien" und "Kommentare". Zur besseren Übersicht sind diese Teilmodule in der Collection "Weblog" zusammengefasst.



Über "CP::Forms > Weblog" können die Teilmodule aufgerufen werden. Für jedes Modul gibt es dort eigene Eingabemasken, über die sich z.B. die Einträge verfassen lassen. Generell gilt dabei, dass nur freigeschaltete Einträge / Kategorien usw. auch innerhalb des Weblogs erscheinen. Es können also Einträge auch schrittweise vorbereitet werden, bevor diese in der finalen Fassung freigeschalten werden.

Grundlegende Einstellungen des Weblogs lassen sich im Modul "Einstellungen" treffen. Dazu gehören u.a. die max. Anzahl der anzuzeigenden Einträge in den einzelnen Feeds oder auch die automatische Aktualisierung der BlogRoll. Weitere Detaileinstellungen können dann direkt durch die einzelnen Parameter der Container festgelegt werden (u.a. die max. Cache-Zeit).



Für fortgeschrittene Anwender gibt es die Möglichkeit, über "Modul bearbeiten" auch die einzelnen Module weiter den eigenen Wünschen anzupassen, indem diese um neue Felder erweitert werden. Hier ist allerdings stets darauf zu achten, dass die allgemeinen Angaben eines Modules, sowie die bisherigen Felder **nicht** verändert werden, da es sonst u.U. zu Problemen kommen kann.

5.7.3. Einbindung des Weblogs auf der Webseite

Alle Inhalte des Weblogs, die über den Administrationsbereich eingegeben werden - also z.B. die Einträge des Weblogs, Kategorien und BlogRoll-Einträge - werden über verschiedene Container auf der Webseite ausgegeben. Jeder dieser Container ist im folgenden genauer erklärt.

Um nun jedoch aus diesen Containern ein komplettes Weblog zusammenzustellen, müssen diese sinnvoll miteinander kombiniert werden. Dazu zuerst ein komplettes Beispiel der Einbindung des Weblogs samt Einträgen, Kommentaren, Kategorien und BlogRoll sowie der Anzeige von Listen- und Detailansichten:

```
<table cellspacing="2" cellpadding="2" width="100%" border="0">
    <tr>
        <td valign="top" width="170">

            <div class="weblog_home"><a title="Startseite" href="/{cp:configuration:main:directory}index.php?id=
{cp:variable:id}">Startseite</a></div>

            {cp:container module="weblog_sections" type="list" order="title" cache="3600"}

            <div class="weblog_line"></div>
```

```
            <p><strong>Letzte Einträge:</strong></p>

            <p>{cp:container module="weblog" type="list" show_date="0"
template="modules/weblog/templates/list_short.html" section="no" cache="3600"} </p>

            <div class="weblog_line"></div>

            <p><strong>Letzte Kommentare:</strong></p>

            <p>{cp:container module="weblog_comments" type="list"
template="modules/weblog_comments/templates/list_short.html" alttext="Noch keine Kommentare vorhanden."
cache="3600"} </p>

            <div class="weblog_line"></div>

            <p><strong>BlogRoll:</strong></p>

            <p>{cp:container module="weblog_blogroll" type="list" cache="3600"}</p>

            <div class="weblog_line"></div><br />

            <div style="padding-bottom: 6px"><a href="/{cp:configuration:main:directory}webservices.php?
file=weblog&amp;type=rss"><img alt="" src="/{cp:configuration:main:directory}modules/weblog/files/icon.rss.gif"
border="0" /></a> <a href="/{cp:configuration:main:directory}webservices.php?file=weblog&amp;type=atom"><img
alt="" src="modules/weblog/files/icon.atom.gif" border="0" /></a></div>

            <div><a href="/{cp:configuration:main:directory}webservices.php?file=weblog&amp;type=rdf"><img alt=""
src="/{cp:configuration:main:directory}modules/weblog/files/icon.rdf.gif" border="0" /></a> <a title="Meine
Linksphäre" href="http://blogstats.de/index.php?url={cp:variable:fullurl}{cp:variable:directory}webservices.php?
file=weblog&type=rss"><img border="0" alt=""
src="http://blogstats.de/images/blogstats_mini_80x15.gif" /></a></div>
        </td>
        <td valign="top">
            {cp:ifnot:oid}
                {cp:container module="weblog" type="list" alttext="Es sind bislang noch keine Einträge vorhanden!"}
            {/cp:ifnot:oid}

            {cp:if:oid}
                {cp:container module="weblog" type="details" number="{cp:variable:oid}" alttext="Es gibt keinen
Eintrag mit dieser ID!"}
            {/cp:if:oid}
        </td>
    </tr>
</table>
```

Dieser Quellcode beinhaltet die einzelnen Containeraufrufe. Im unteren Teil wird dazu mittels der {cp:tags} "{cp:if}" und "{cp:ifnot}" unterschieden, ob sich der Besucher auf einer Detailseite (Parameter "oid" ist in der URL vorhanden) oder auf der Startseite bzw. innerhalb einer Weblog-Kategorie befindet (Parameter "oid" ist nicht vorhanden.

Um den Weblog in dieser Form einzubinden, ist zuerst ein neuer Artikel im Administrationsbereich anzulegen. Dort wird auf den HTML-Modus gewechselt und der obenstehende Quellcode wird per Copy&Paste übernommen. Anschließend kann nach dem Speichern bereits auf den neuen Weblog gewechselt werden.

An der Vorlage "main.htm" aus dem Vorlagenset ist zusätzlich noch eine Anpassung vorzunehmen, damit die CSS-Befehle zur Gestaltung des Weblogs verarbeitet werden können. Folgende Zeile ist dort in den -Bereich einzubinden:

```
<link rel="stylesheet" href="/{cp:configuration:main:directory}modules/weblog/templates/weblog.css" type="text/css"
media="screen, print" />
```

Ausgehend von diesem Quellcode und den einzelnen, darauf abgestimmten Vorlagen in den Ordnern unter "/modules/weblog/templates/" und "/modules/weblog_*/templates/" kann nun unter Einsatz der einzelnen Container das Weblog weiter modifiziert werden, um beispielweise gesonderte Seiten für eine Übersicht über die letzten Kommentare anzuzeigen, weitere Grafiken einzubauen etc.

5.7.4. Anpassung der Vorlagen für die Einträge, Kategorien, BlogRoll usw.

Für die Darstellung der einzelnen Elemente des Weblogs werden zahlreiche Vorlagen genutzt, welche eine starke Differenzierung der Gestaltung ohne weitere Programmierung ermöglichen. Diese Vorlagen sind in den folgenden Ordnern vorzufinden:

- **"/modules/weblog/templates/"**
  Vorlagen für die Einträge des Weblogs.
- **"/modules/weblog_blogroll/templates/"**
  Vorlagen für die Einträge der BlogRoll.
- **"/modules/weblog_comments/templates/"**
  Vorlagen für die Kommentare.
- **"/modules/weblog_options/templates/"**
  Vorlagen für die Einstellungen (nicht im öffentlichen Bereich genutzt).
- **"/modules/weblog_sections/templates/"**
  Vorlagen für die Kategorien.

Eine genauere Auflistung der genutzten Vorlagen kann innerhalb des Administrationsbereiches über "CP::Forms > Weblog > [MODUL] > Vorlagen-Erklärung" abgerufen werden, wie z.B. über "admin.php?file=forms&mode=show_templateinfo&form[module]=weblog" für die Einträge des Weblogs. Zusätzlich werden in der

sogenannten "{cp:tags}-Erklärung" die jeweils innerhalb der Vorlagen zur Verfügung stehenden {cp:tags} aufgelistet.

5.7.5. Container: Anzeige der Einträge des Weblogs

Dieser Container vom Typ "list" dient zur Anzeige der im Weblog enthaltenen und freigeschalteten Einträge. Er ist Bestandteil der normalen Einbindung des Weblogs auf der Webseite, kann aber auch davon unabhängig auf einer beliebigen Seite in Contentpapst platziert werden.

```
{cp:container module="weblog" type="list"} // normale Auflistung der Übersicht mit Gruppierung nach dem Datum
```

Über den Parameter "template" kann eine alternativ zu verwendende Vorlage festgelegt werden. Die Informationen zu den standardmäßig genutzten Vorlagen / verfügbaren {cp:tags} sind dabei unter "CP::Forms > Weblog > Einträge" und dann "Vorlagen-Erklärung" / "{cp:tags}-Erklärung" abrufbar.

Über den Parameter "show_date" kann angegeben werden, ob nach dem Datum gruppiert und das Datum oberhalb der Einträge an diesen Tag ausgegeben werden soll. Steht dieser Parameter auf "0", dann wird eine einfache Liste generiert. Dies ist z.B. sinnvoll, wenn neben der eigentlichen Auflistung der - oft sehr umfangreichen - Einträge noch eine kurze Auflistung der Titel generiert werden soll (in Kombination mit dem Parameter "template" für eine alternative Vorlage):

```
{cp:container module="weblog" type="list" show_date="0" template="modules/weblog/templates/list_short.html"}
```

Folgende Parameter sind für diesen Container zulässig / erforderlich:

Parameter für diesen Container		
dir	Es wird aufsteigend ("asc") oder absteigend ("desc") geordnet.	optional
limit	Anzahl der insgesamt angezeigten Einträge.	optional
section	Mit dem Wert "no" werden die Einträge unabhängig von der Kategorie angezeigt.	optional
show_date	Datum der Einträge anzeigen.	optional
template	Vorlage, welche für die Anzeige verwendet wird.	optional
type	Für diesen Container muss als Wert "list" verwendet werden.	erforderlich

5.7.6. Container: Anzeige anderer Weblogs über die BlogRoll

Über die sogenannte "BlogRoll" können andere empfehlenswerte Weblogs auf der Webseite eingeblendet werden. Im Unterschied zu einer normalen Liste beinhaltet die BlogRoll jedoch noch ein ganz besonderes Feature: Die Anzeige erfolgt nach dem Datum der letzten Änderung dieser Weblogs.

Dazu werden im Administrationsbereich zuerst die gewünschten Weblogs erfasst, samt der URL's zu den jeweiligen Feeds dieser Weblogs. CP::Weblog liest diese Feeds dann im Hintergrund automatisch aus und ermittelt das letzte Änderungsdatum.

Die Anzeige erfolgt dann auch hier wieder über einen Container vom Typ "list":

```
{cp:container module="weblog_blogroll" type="list"} // Auflistung der Weblogs mit Ordnung nach dem Aktualisierungsdatum
```

Über den Parameter "template" kann eine alternativ zu verwendende Vorlage festgelegt werden. Die Informationen zu den standardmäßig genutzten Vorlagen / verfügbaren {cp:tags} sind dabei unter "CP::Forms > Weblog > BlogRoll" und dann "Vorlagen-Erklärung" / "{cp:tags}-Erklärung" abrufbar.

Folgende Parameter sind für diesen Container zulässig / erforderlich:

Parameter für diesen Container		
author	ID des Benutzers, um nur dessen Einträge anzuzeigen.	optional
limit	Anzahl der insgesamt angezeigten Einträge.	optional
template	Vorlage, welche für die Anzeige verwendet wird.	optional
type	Für diesen Container muss als Wert "list" verwendet werden.	erforderlich

5.7.7. Container: Auflistung der Kategorien des Weblogs

Auch die Auflistung der einzelnen Kategorien des Weblogs auf der Webseite erfolgt über einen Container vom Typ "list":

```
{cp:container module="weblog_sections" type="list" order="title"} // Ausgabe der Kategorien nach alphabetischer Sortierung
```

Über den Parameter "template" kann eine alternativ zu verwendende Vorlage festgelegt werden. Die Informationen zu den standardmäßig genutzten Vorlagen / verfügbaren {cp:tags} sind dabei unter "CP::Forms > Weblog > Kategorien" und dann "Vorlagen-Erklärung" / "{cp:tags}-Erklärung" abrufbar.

Folgende Parameter sind für diesen Container zulässig / erforderlich:

Parameter für diesen Container		
dir	Es wird aufsteigend ("asc") oder absteigend ("desc") geordnet.	optional
limit	Anzahl der insgesamt angezeigten Einträge.	optional
order	Sortierung der Artikel (z.B. "number", "title", "description", "date_added")	optional

template	Vorlage, welche für die Anzeige verwendet wird.	optional
type	Für diesen Container muss als Wert "list" verwendet werden.	erforderlich

5.7.8. Container: Auflistung der Kommentare zu einem Eintrag

CP::Weblog bietet über die Trackbacks bzw. das normale Kommentar-System den Besuchern die Möglichkeit, die Weblog-Einträge zu kommentieren. Die Anzeige des Formulares zum Eintragen der Kommentare, als auch die Ausgabe der Kommentare selbst, erfolgt über Container vom Typ "list" und "form":

```
{cp:container module="weblog_comments" type="form" entry="4"} // Formular für den Weblog-Eintrag "4"
```

Der obenstehende {cp:tag} bewirkt die Anzeige des Formulares zum Eintragen der Kommentare speziell für den Weblog-Eintrag mit der ID "4". D.h. die Kommentare werden intern bereits diesem Eintrag zugeordnet. Um diese nun auch wieder ausgeben zu lassen, ist folgendermaßen vorzugehen:

```
{cp:container module="weblog_comments" type="list" entry="4"} // Anzeige der Kommentare für den Weblog-Eintrag "4"
```

Mit diesem {cp:tag} werden die vorher eingegebenen Weblog-Einträge wieder angezeigt über die Vorlage "/modules/weblog_comments/list_public.html". Über den Parameter "template" kann eine alternativ zu verwendende Vorlage festgelegt werden. Die Informationen zu den standardmäßig genutzten Vorlagen / verfügbaren {cp:tags} sind dabei unter "CP::Forms > Weblog > Kommentare" und dann "Vorlagen-Erklärung" / "{cp:tags}-Erklärung" abrufbar.

Speziell für die Generierung einer Übersicht über die allgemein zuletzt zum Weblog hinzugefügten Kommentare der Benutzer kann der folgende {cp:tag} genutzt werden:

```
{cp:container module="weblog_comments" type="list" template="alternativ.html"}
```

Folgende Parameter sind für diesen Container zulässig / erforderlich:

Parameter für diesen Container		
dir	Es wird aufsteigend ("asc") oder absteigend ("desc") geordnet.	optional
entry	Beschränkung auf einen bestimmten Weblog-Eintrag (ID eingeben).	optional
limit	Anzahl der insgesamt angezeigten Einträge.	optional
order	Sortierung der Artikel (z.B. "number", "title", "description", "date_added")	optional
template	Vorlage, welche für die Anzeige verwendet wird.	optional
type	Für diesen Container muss als Wert "list" oder "form" verwendet werden.	erforderlich

5.7.9. Export der Einträge als RSS-, ATOM-Feed usw.

CP::Weblog stellt die Einträge des Weblogs automatisch in verschiedenen Export-Formaten zum Abruf bereit. U.a. wird ein Export als RSS- und ATOM-Feed angeboten. Dies ermöglicht es anderen Benutzern z.B. den Weblog auf der eigenen Webseite einzubinden oder über einen Feed-Reader immer auf dem aktuellsten Stand zu bleiben. Es werden jeweils nur freigeschaltete Einträge exportiert, deren Veröffentlichungsdatum <= dem aktuellen Datum ist.

URL's für den Export der Einträge	
webservices.php?file=weblog...	ATOM-Format (Version 0.3)
webservices.php?file=weblog...	RDF-Format (Version 1.0)
webservices.php?file=weblog...	RSS-Format (Version 2.0)

5.8. Dateiverwaltung

Funktionen zur Verwaltung der Dateien Ihrer Webseite stehen Ihnen in diesem Modul zur Verfügung. Es können alle aus dem Desktop-Bereich bekannten Operationen durchgeführt werden, wie z.B. die Erstellung und Verwaltung von Dateien und Ordnern. Zusätzlich kann für HTML-Dateien ein optionaler WYSIWYG-Editor genutzt werden.

5.8.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=upload	Aufruf über den Administrationsbereich

5.8.2. Datei erstellen

Unabhängig von News und Artikeln können mit dem WYSIWYG-Editor oder per einfacher Eingabefelder komplett neue Dateien angelegt werden. Diese verwenden keine Vorlagen und können dementsprechend für Startseiten mit Flashanimationen usw. genutzt werden. Andererseits lassen sich auch ganze PHP- oder JS-Scripte bzw. XML-Dateien anlegen.

Um eine neue Datei zu erstellen, wählen Sie zuerst im Dateibrowser den gewünschten Ordner aus und geben Sie dann in der Box "Datei erstellen" einen Dateinamen sowie das gewünschte Dateiformat an. Wenn Sie eine HTML-Datei erstellen möchten, wird daraufhin der bereits aus dem Artikel-Bereich bekannte WYSIWYG-Editor eingeblendet. Vorlagen für Artikel können hier einmalig genutzt werden. Bei späterer Anpassung der Dateien ist jedoch eine einfache Umstellung auf andere

Vorlagen nicht möglich.

Bei PHP-, JS- oder XML-Dateien wird hingegen ein einfaches Eingabeformular eingeblendet. Hier kann der Quelltext eingegeben werden, wie er später ausgeführt werden soll. Bei beiden Editoren wird die neue Datei durch einen Klick auf "Abschicken" an der vorher bestimmten Position im Dateisystem angelegt.

5.8.3. Datei kopieren

Kopien von Dateien werden in Contentpapst mittels der Checkbox vor dem Dateinamen angelegt. Klicken Sie diese Checkbox an und gehen Sie anschließend auf "Aktion auswählen > Kopieren". Im gleichen Ordner wird daraufhin eine Kopie angelegt mit dem Namen "Kopie_von_*" (* entspricht dem alten Dateinamen). Durch die Auswahl mehrerer bzw. aller Dateien ("Alle markieren") lässt sich der Vorgang gesammelt durchführen.

5.8.4. Datei löschen

Wählen Sie in der Zeile der entsprechenden Datei das Icon an, wird die Datei aus der Dateiverwaltung gelöscht. Für mehrere Dateien kann dieser Vorgang durch einen Klick auf "Alle markieren" beschleunigen. Gelöschte Dateien können nicht wiederhergestellt werden.

5.8.5. Datei umbenennen

Wählen Sie in der Zeile der entsprechenden Datei das Icon an, wird eine Seite aufgerufen, in welcher ein neuer Name der Datei angegeben werden kann. Nach dem Klick auf "Abschicken" wird der Dateiname geändert. Bestehende Verlinkungen zu dieser Datei in Artikeln etc. müssen jedoch ebenfalls geändert werden.

5.8.6. Ordner umbenennen

Wählen Sie in der Zeile des entsprechenden Ordners das Icon an, wird eine Seite aufgerufen, in welcher ein neuer Name für den Ordner angegeben werden kann. Nach dem Klick auf "Abschicken" wird der Name des Ordners geändert. Bestehende Verlinkungen zu Dateien innerhalb des Ordners müssen jedoch ebenfalls geändert werden.

5.8.7. Zugriffssperre

Um sicherzustellen, dass Arbeiten an den Dateien nicht behindert werden, wird der Zugriff auf die Dateien in der Datenbank gespeichert. Greift ein zweiter Redakteur auf eine bereits durch einen anderen Redakteur geöffnete Datei zu, so wird diesem der Zugriff verwehrt. Ebenso wird bei der Löschung und Umbenennung von Dateien verfahren. Der Zugriff auf die Datei wird entweder durch die Speicherung wieder freigegeben oder durch den Ablauf einer festgelegten Zeitspanne.

Bei Contentpapst sind dies standardmäßig 30min. D.h. wenn nach dieser Zeit noch keine Speicherung erfolgt ist, wird die Datei dennoch wieder zur Bearbeitung für andere Redakteure freigegeben. Durch diese Regelung kann die Redaktionsarbeit ohne Behinderungen und bei gleichzeitiger Sicherung der Dateien erfolgen. Sollte eine schnellere Freigabe nötig sein, haben Sie immer die Möglichkeit, mit dem zuständigen Redakteur Kontakt aufzunehmen.

5.9. Einstellungen

Beinahe alle Einstellungen des Systems können nach der Installation direkt über den Administrations-Bereich durch die berechtigten Nutzer angepasst werden. Dazu gehören z.B. Standard-Metatags, Cache und Logging.

5.9.1. Allgemeine Einstellungen

In den "Allgemeinen Einstellungen" finden Sie viele elementare Einstellungen, die für den Betrieb des Systems wichtig sind. Dazu gehören u.a. die Einstellungen der richtigen URL's und einer Standardsprache. Viele dieser Einstellungen sind direkt auf dieser Seite erklärt. Einige Einstellungen seien aber zusätzlich hier erwähnt:

Titel der Webseite	Seitenname, z.B. die Firmierung Ihres Unternehmens (bspw. "MaxMustermann KG")
Domain der Webseite	Domain, auf der die Webseite liegt (z.B. "http://www.meinedomain.de/")
Pfad zum CMS	Der Pfad ausgehend von der Domain (bei "http://www.meinedomain.de/cp/" ist es nur "cp/")
GZIP-Komprimierung	Stärke der Komprimierung der Inhalte der Webseite vor der Auslieferung an den Browser
Datum	Formatierung des Datums auf der Webseite
Datum (Kurzform)	Formatierung des Datums auf der Webseite
Standardsprache	Auswahl der Standardsprache aus den für das System verfügbaren Sprachen
Zeichensatz	Verwendeter Zeichensatz (wird je nach Sprache automatisch gesetzt)
WYSIWYG-Editor	Auswahl des gewünschten WYSIWYG-Editors (interner Editor oder FCK Editor)

Wichtige URL's für dieses Modul	
admin.php?file=options	Aufruf über den Administrationsbereich

5.9.2. Cache

Contentpapst speichert viele Inhalte nach dem einmaligen Abruf in einem Cache. D.h. sobald z.B. ein Artikel durch einen Besucher aufgerufen wird und dieser noch nicht im Cache vorhanden ist, generiert Contentpapst automatisch eine Kopie, die im Cache auf dem Server abgelegt und an den Besucher ausgeliefert wird. Weitere Besucher erhalten bis zum Ablauf der Cache-Zeit automatisch die Cache-Datei. Dies spart Rechenzeit und erhöht die Performance Ihres Webservers erheblich.

Der Cache ist in Contentpapst u.a. für Module wie die Artikel standardmäßig vorhanden und kann dort jeweils einzeln gesetzt werden. Alle Angaben sind dabei in Sekunden zu machen (für einen Tag bspw. "86400" Sekunden). Bei "0" Sekunden bzw. ohne Auswahl des Caches bleibt dieser deaktiviert. Dies kann beispielsweise bei Inhalten sinnvoll sein, die dynamische Elemente wie z.B. PHP beinhalten. Als Besonderheit führt Contentpapst im Hintergrund bei allen Aktionen automatische Aktualisierungen des Caches durch. Daher erscheinen Inhalte immer in der neuesten Fassung auf der Webseite, ohne dass erst der Ablauf der regulären Cache-Zeit abgewartet werden muss. Eine weitere Besonderheit ist der individuelle Cache für jeden einzelnen Container. Hier kann über den Parameter "cache" stets eine Cache-Zeit in Sekunden angegeben werden.

In den Einstellungen haben Sie die Möglichkeit, den Cache insgesamt zurückzusetzen. Dann wird der gesamte Inhalt des Ordners "cache" entfernt. Dies kann sinnvoll sein, wenn Änderungen an genutzten Vorlagen vorgenommen wurden etc.

5.9.3. Sitemaps für Google, MSN und Yahoo!

Die Suchmaschine Google ermöglicht Seitenbetreibern unter dem Namen "Google Sitemaps" die Übermittelung ganzer Sitemaps zu schnelleren und gezielteren Indexierung der in einer Webseite enthaltenen Seiten. Contentpapst kann diese Sitemaps für das Artikel-, News- und (sofern vorhanden) das Shop-Modul automatisch generieren. In den Einstellungen kann unter "Sitemaps für Google, MSN und Yahoo!" die Generierung dieser Sitemaps gestartet werden. Wahlweise kann das System Google im Anschluss daran auch direkt den Speicherort dieser Datei übermitteln.

Wenn Sie den genauen Stand der Indexierung überprüfen oder Sitemaps zusätzlich manuell hinzufügen wollen, dann können Sie sich unter "https://www.google.com/webmasters/sitemaps/stats" für Google Sitemaps anmelden. Zusätzlich können diese Sitemaps auch bei MSN und Yahoo! genutzt werden.

5.9.4. Kontakt

Auf dieser Seite der Einstellungen müssen lediglich 3 Email-Adressen angegeben werden:

Zentrale Email-Adresse	z.B. "info@ihre-domain.de"
Email-Adresse des Webmasters	z.B. "webmaster@ihre-domain.de"
Email-Adresse für Support	z.B. "support@ihre-domain.de"

Diese Email-Adressen werden in verschiedenen Bereichen des Systems benötigt. Sie werden immer dann eingesetzt, wenn keine alternative Email-Adresse eingegeben wurde, z.B. bei der Generierung neuer (Kontakt-)Formulare in CP::Forms oder beim Versand von Anfragen über den Kundenservice.

5.9.5. Logging

Contentpapst beobachtet im Hintergrund die Aktionen der eingeloggten Benutzer inkl. der Redakteure und des Administrators. Aktionen wie das Hinzufügen und Bearbeiten eines Artikels werden dabei mitgeloggt, d.h. sie werden mit dem Datum und weiteren Angaben in der Datenbank gespeichert. In den Einstellungen kann unter "Logging" dann eine Übersicht der bislang mitgeloggten Aktionen angezeigt werden. Die Anzeige kann dabei nach der Art der Aktion und nach dem Zeitraum eingegrenzt werden. Auch gescheiterte Login-Versuche etc. werden hier gespeichert, so dass sich eventuelle Zugriffe durch Dritte leichter erkennen lassen.

Zusätzlich zu dieser Übersicht kann das Logging in den Einstellungen noch genauer konfiguriert werden, denn neben den Aktionen der Benutzer können wahlweise auch PHP-Fehlermeldungen und -Warnungen mitgeloggt werden. Darauf aufbauend kann Contentpapst dann bei besonders kritischen Fehler oder einem unerlaubten Zugriff durch Dritte z.B. eine Email an den Administrator versenden.



Das Logging von PHP-Fehlermeldungen und -Warnungen ist nur möglich, wenn der auf dem Server für PHP eingestellte Error-Level dem Standard-Wert "error_reporting = E_ALL & ~E_NOTICE & ~E_STRICT" entspricht. Sollte dies nicht der Fall sein, dann werden u.U. auch Notizen etc. mitgeloggt, was die Funktionsfähigkeit des Systems einschränken kann.

5.9.6. Metatags

Contentpapst unterstützt zur Optimierung der Seite für Suchmaschienen etc. auch die Vergabe von individuellen Metatags. Voraussetzung ist, dass in der Vorlage "main.htm" des genutzten Vorlagen-Sets {cp:tags} im Format "{cp:meta:*}" genutzt werden. In den Einstellungen können allgemeingültige Metatags angeben werden. Diese werden immer dann zum Einsatz gebracht, wenn es für den jeweils angezeigten Inhalt keine eigenen Metatags gibt. Folgende {cp:tags} werden für die Metatags unterstützt.

{cp:meta:title}	Titel der Webseite (z.B. "Luftfahrtvereinigung Musterhausen")
{cp:meta:description}	Beschreibung der Webseite
{cp:meta:keywords}	Schlüsselwörter der Webseite (z.B. "Piloten, Luftfahrt, Flugzeuge, Flughafen...")
{cp:meta:copyright}	Name und Email-Adresse der für die Umsetzung verantwortlichen Person / Unternehmung
{cp:meta:author}	Name und Email-Adresse der für die Inhalte verantwortlichen Person / Unternehmung
{cp:meta:publisher}	Name und Email-Adresse der für die Veröffentlichung verantwortlichen Person / Unternehmung
{cp:meta:robots}	Angaben zum Indexierungsverhalten von Suchmaschinen

Weiterhin gibt es auf dieser Seite den Abschnitt "Neue Metatags erstellen" über den Sie den HTML-Code für die Metatags erstellen können. Hier können Sie alternativ direkt Texte eingegeben. Wenn Sie diesen HTML-Code jedoch im Vorlagen-Set verwenden, findet ein Austausch der Metatags je nach Seite nicht mehr statt.

Neben diesen allgemeinen Metatags gibt es dann für Module wie die Artikel jeweils eigene Eingabefelder. So können Sie beispielsweise einem Artikel gezielt die passenden Schlüsselwörter mitgeben und so Ihre Seite weiter optimieren.

5.9.7. Modulaktivierung

Die einzelnen Module von Contentpapst lassen sich auch nachträglich aktivieren, löschen oder verstecken usw. Auf der Seite "Modulaktivierung" in den Einstellungen sehen Sie eine Übersicht aller derzeit vorhandenen Module. Vorhandene Module können auf "Nicht aktiv" (nur im Administrations-Bereich sichtbar) und "Verstecken" (nicht sichtbar) gesetzt werden.

Sobald Sie ein neues Modul erhalten und dieses im Ordner "/modules/" abgelegt haben, sehen Sie in dieser Übersicht einen neuen Eintrag samt einem Button mit der Beschriftung "Installation". Durch einen Klick auf diesen Button werden die notwendigen Datenbankinhalte zur Datenbank hinzugefügt und der Status des Modules wird auf "Aktiv" geändert.

5.9.8. Server

Auf der Seite "Server" finden Sie in den Einstellungen ausführliche Informationen über die Konfiguration Ihres Servers bzw. zu den darauf laufenden Systemen PHP, MySQL und Apache. U.a. erfahren Sie hier mehr über die derzeit genutzten Versionen von PHP und Apache, die Server-Zeit usw. Diese Informationen können durch den Kundensupport angefordert werden, wenn Sie Probleme mit der Installation oder dem Betrieb des Systems haben.

5.9.9. Suchmaschinenfreundliche URL's

Alle Seiten werden bei Contentpapst dynamisch bzw. semi-dynamisch erzeugt und können dann temporär in einem Cache zwischengespeichert werden. Dies ermöglicht einerseits die Darstellung einer stets aktuellen Seite, gleichzeitig kann jedoch durch den Cache das Lastaufkommen drastisch gesenkt werden.

Diese dynamischen Seiten sind meist über URL's mit angehängten Parametern erreichbar, mit denen Contentpapst mitgeteilt wird, beispielsweise welche Seite eines Artikels gerade gelesen wird. Diese URL's (z.B. "index.php?file=article&mode=detail&number=32&page=5") können problematisch sein, wenn es um die Aufnahme in Suchmaschinen geht. Einige Suchmaschinen indizieren dynamische Seiten aus verschiedenen Gründen grundsätzlich nicht.

Für Contentpapst gibt es allerdings eine elegante Lösung, um diese Blockade zu umgehen. Mit Hilfe der Option "Suchmaschinenfreundliche URL's" in den Einstellungen, können URL's auf ein normales HTML-Format gebracht werden. Hierzu werden Veränderungen in der Datei ".htaccess" vorgenommen. Die weiter oben als Beispiel genannte Artikel-Seite wäre mit aktivierter Option z.B. auch unter "/artikelkategorie/titel-des-artikels/" abrufbar. Für Suchmaschinen und den Besucher augenscheinlich eine normale HTML-Seite, intern jedoch weiterhin PHP.

Um bei einem Apache-Webserver diese Option nutzen zu können, muss die Erweiterung "mod_rewrite" installiert sein. Für den Microsoft IIS-Webserver wird ebenfalls ein Zusatztool benötigt, hier kann unter anderem IIS Rewrite zum Einsatz kommen. Für den IIS müssen die in der ".htaccess" enthaltenen Regeln in einer Datei "rewrite.ini" abgelegt und ins Verzeichnis "%windir%system32inetsrv/IISRewrite" verschoben werden. Im letzteren Fall sind zusätzlich absolute statt relative Pfade anzugeben.

Die Suchmaschinenfreundlichen URL's können Sie direkt in den Einstellungen aktivieren. Die Datei ".htaccess" wird dann automatisch aktualisiert. Achten Sie aber bitte vorher darauf, dass dieser Datei per FTP ein ausreichender CHMOD (775 bzw. 777) gegeben wird, damit eine Bearbeitung durch Contentpapst möglich ist.

5.9.10. Volltextsuche

Über die Volltext-Suche unter "index.php?file=search" können die Besucher gleichzeitig nach Inhalten im Artikel-, News-, dem Shop-Modul (sofern vorhanden) sowie in allen für die Volltextsuche freigeschalteten CP::Forms-Modulen suchen. Alle grundlegenden Einstellungen für die Volltextsuche können in den Einstellungen auf der gleichnamigen Seite vorgenommen werden. Dazu gehören u.a. eine Farbmarkierung für die Suchworte in den Suchergebnissen (vergleichbar mit den Google-Suchergebnissen), die Eingrenzung der Länge der Suchworte und diverse Filter, die bei der Suche genutzt werden können.

5.9.11. Vorlagen-Sets

Der Aufbau und Einsatz von Vorlagen in den verschiedenen Bereichen des Systems wird im Detail im Abschnitt "Gestaltung von Vorlagen" erläutert. Dort finden Sie u.a. nähere Informationen zu den "Vorlagen-Sets", in denen die einzelnen Vorlagen eines Mandanten, d.h. einer Webseite, zusammengefasst werden. In den Einstellungen kann unter "Mandantenverwaltung" das standardmäßige Vorlagen-Sets festgelegt werden, ebenso kann hier ein komplett neuer Mandant auf der Basis des derzeit aktivierten Mandanten angelegt werden. Alle Dateien und Ordner des entsprechenden Vorlagen-Sets werden dazu in einen neuen Ordner mit dem Namen "/themes/__NAME_DES_MANDANTEN__/" kopiert.



Bitte achten Sie beim Anlegen eines neuen Mandanten darauf, dass Sie vorher per FTP für den Ordner "/themes/" die Zugriffsrechte ändern, damit Contentpapst in diesem Ordner neue Ordner und Dateien anlegen kann. Für diesen Ordner ist der CHMOD 777 bzw. 775 erforderlich.

5.10. Galerie

Zeigen Sie Bilder von Ausstellungen, Ihren Produkten etc. Die Möglichkeiten des Einsatzes der Galerie sind äußerst vielfältig. Auf Wunsch kann die Galerie durch eine Anpassung der Vorlagen auch für normale Besucher freigeschalten und von diesen erweitert werden. Nutzen Sie die Vorteile der grafischen Präsentation.

5.10.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=gallery	Aufruf über den Administrationsbereich

index.php?file=gallery	Aufruf über den öffentlichen Bereich
gallery.html	Aufruf über den öffentlichen Bereich (Suchmaschinenfreundliche URL's)

5.10.2. Mitglieder-Galerien

Die Anzeige der Bilder in der Galerie kann auch auf einzelne Mitglieder beschränkt werden. Hängen Sie hierzu beim Aufruf der Galerie die Variable "poster" mit der jeweiligen Mitglieder-Nummer an. Besitzt ein Mitglied z.B. die Nummer "1", so können dessen Bilder in der Galerie unter "index.php?file=gallery&poster=1" eingesehen werden.

Eine Kombination mit den Variablen für die Ordnung der Galerie nach best. Kriterien ("picname" [Bildname], "time" [Zeit], "count" [Anzahl der Aufrufe], "avrating" [Bewertung]), der Ordnungs-Richtung ("asc" [aufsteigend], "desc" [absteigend]) und der Kategorie ist weiterhin möglich.

Anwendungsbeispiele:

```
index.php?file=gallery&poster=1&order=picname&dir=asc
// Bilder der Mitglieds mit der Nummer "1" werden alphabetisch aufsteigend nach dem Bildnamen geordnet
```

```
index.php?file=gallery&poster=19&order=time&dir=desc&cat=2
// Bilder des Mitglieds mit der Nummer "19" aus der Kategorie mit der Nummer "2" werden absteigend nach der Zeit der Einsendung angezeigt (neueste Bilder zuerst).
```

5.11. Gästebuch

Das Gästebuch bildet im Gegensatz zum Kontakt-Formular die Eingaben Ihrer Besucher direkt ab. Somit bleiben diese auch für andere Besucher Ihrer Webseite sichtbar. Mittels umfangreicher Einstellungen können Sie das Gästebuch nach Ihren Wünschen anpassen.

5.11.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=guestbook	Aufruf über den Administrationsbereich
index.php?file=guestbook	Aufruf über den öffentlichen Bereich
guestbook.html	Aufruf über den öffentlichen Bereich (Suchmaschinenfreundliche URL's)

5.11.2. Einstellungen

Einzelne Eingabefelder können auf der Webseite angezeigt/gesperrt werden. Dazu gehören insbesondere die Anzeige von Email-Adresse, Herkunft, Homepage und Datum des Eintrages. Die Anzeige der IP-Adresse ist ebenfalls möglich. Orientieren Sie sich hierbei jedoch bitte an den jeweils gültigen Datenschutzrichtlinien.

Weitere Einstellungen betreffen die Einschränkungen der Länge der Einträge, sowie die Einträge pro Seite. Um den Aufbau des Layouts und die Sicherheit der Webseite nicht zu gefährden, können HTML und Scripte in Einträgen unterbunden werden. Zusätzlich können Funktionen zur Blockierung doppelter Einträge aktiviert werden.

Wenn Sie über neue Einträge informiert werden wollen, kann mit der Funktion "Benachrichtigung bei neuen Einträgen" automatisch eine Email an die standardmäßig eingestellte Email-Adresse gesendet werden. Um sicherzugehen, dass keine unerwünschten Einträge auf der Webseite erscheinen, lassen sich diese auch manuell freischalten.

5.11.3. Emoticons / Smilies

Emoticons, oder auch Smilies sind eine Möglichkeit für Ihre Besucher, ihren Gästebucheinträgen ein wenig mehr Ausdrucksstärke zu verleihen. Ist die Funktion aktiviert, werden alle Grafiken aus dem Ordner "/images/smilies/" im Formular zum Erstellen eines neuen Eintrages angezeigt. Achten Sie darauf, dass keine zu großen oder zu ladeintensiven Bilder in diesem Ordner hinterlegt werden, da dies sonst die Darstellung der Beiträge eventuell negativ beeinflussen würde.

5.11.4. Vorschau anzeigen

Wenn diese Funktion aktiviert ist, lassen sich Einträge vorab in einer Vorschau anzeigen. So kann der Besucher noch einmal die Gestaltung sowie den Inhalt des Eintrages kontrollieren, bevor er diesen letztendlich absendet.

5.12. Kalender (Veranstaltungen / Termine)

*Das Kalender-Modul ist ein **kostenloses Zusatzmodul** für **Contentpapst**. Über dieses Modul können Veranstaltungen und sonstige Termine aus beliebigen CP::Forms-Modulen über einen gemeinsamen Kalender angezeigt und miteinander verknüpft werden. Ausgehend von den einzelnen Terminen kann dann zu den detaillierten Einträgen gewechselt werden. Das Kalender-Modul kann unter "Module > CP::Forms > Modulvorlagen" heruntergeladen werden.*

5.12.1. Installation des Kalender-Modules

Das Kalender-Modul wird zusätzlich zu einer bestehenden Contentpapst-Installation als CP::Forms-Modul installiert. Dazu wird das Modul unter "Module > CP::Forms > Modulvorlagen" als Modulvorlage für die Erstellung eines neuen Modules

bereitgestellt. Über "Modul aus Vorlage erstellen" legen Sie dann ein neues Modul auf der Basis des Kalender-Modules an. Sobald das Modul erstellt wurde, kann dieses über den Administrationsbereich unter "Module > CP::Forms" aufgerufen werden.

5.12.2. Benutzung des Kalender-Modules im Administrationsbereich





Das auf CP::Forms-basierende Kalender-Modul ermöglicht die gemeinsame Darstellung von Veranstaltungen und Terminen beliebiger CP::Forms-Module in Listen- und natürlich auch in Kalender-Form.

Dabei werden innerhalb des Kalender-Modules keine Inhalte eingetragen, sondern das Modul dient lediglich der Verknüpfung der in anderen anderen Modulen eingetragenen Inhalte. Damit diese Verknüpfung zu den Modulen hergestellt werden kann, müssen für jedes Modul, dessen Einträge innerhalb des Kalenders oder der Listen angezeigt werden soll, die folgenden Daten angegeben werden:

- **Ordnername des CP::Forms-Modules**
  Der Ordner-Name des Modules, welches in den Kalender eingebunden werden soll (z.B. "forms_events"). Eine Liste der möglichen Ordner-Namen wird dabei zur Auswahl bereitgestellt, so dass das gewünschte Modul per Klick übernommen werden kann.

- **Feld für Startdatum**
  Der interne Name des Feldes, welches in diesem Modul das Startdatum enthält. Dieses Feld muss vom Typ "Datum" oder "Datum/Zeit" sein.

- **Feld für Enddatum (optional)**
  Der interne Name des Feldes, welches in diesem Modul das Enddatum enthält. Dieses Feld muss vom Typ "Datum" oder "Datum/Zeit" sein.

- **Vorlage für die Anzeige**
  Der komplette Pfad zur Vorlage, welche für die Anzeige in der Auflistung genutzt werden soll. Es können in dieser Vorlage alle {cp:tags} der normalen Listenansicht des jeweiligen Modules genutzt werden.

- **URL für die Detailseite (derzeit optional)**

Sobald diese Daten für ein Modul angegeben wurden, werden automatisch die Einträge des Modules in den Kalender und die Auflistungen mit einbezogen. Weitere Angaben sind hierzu im Administrationsbereich nicht notwendig.

5.12.3. Einbindung des Kalenders auf der Webseite

Für das Kalender-Modul gibt es insgesamt 2 Container, die in den Einträgen "Container: Tagesauflistung der Einträge generieren" und "Container: Kalender auf der Webseite anzeigen" detailliert beschrieben werden. Über die Container können eine Listen- und eine Kalenderansicht an beliebigen Stellen der Webseite integriert werden. Die {cp:tags} können dabei sowohl in einer beliebigen Vorlage, als auch in einem Artikel integriert werden. Auch eine mehrfache Einbindung des Kalenders ist machbar. Eine mögliche Form der Einbindung des Kalenders soll im Folgenden kurz geschildert werden:

In dem mehrspaltigen Layout einer Webseite soll in der rechten Spalte ein Kalender integriert werden. Dazu ist wie im Eintrag "Container: Kalender auf der Webseite anzeigen" beschrieben ein {cp:tag} in der für das Layout genutzten Vorlage an der gewünschten Position zu platzieren. In der Regel ist dies die Vorlage "main.htm" im aktuellen Vorlagen-Set.

Sobald dieser Container platziert wurde und über den Admin-Bereich andere CP::Forms-Module eingebunden wurden, wird dann bereits ein Kalender angezeigt und es kann zwischen den Monaten über eine Navigation gewechselt werden.

Damit durch einen Klick auf einen der Tage dann auch die passenden Einträge dieses Tages angezeigt werden, muss zuerst noch ein neuer Artikel in Contentpapst angelegt werden, welcher mit dem Container für die Listenansicht (wie unter "Container: Tagesauflistung der Einträge generieren" beschrieben) gefüllt wird. Die ID des neuen Artikels muss dem ersten Container dann noch über den Parameter "list_id" mitgeteilt werden.

Der Container für den Kalender könnte also z.B. folgendermaßen aufgebaut sein:

```
{cp:container module="[MODULNAME]" type="show_month" cell_padding="0" cell_spacing="1" border_width="0px"
color_font="#E1E1E1" color_font_title="#E1E1E1" list_id="17" table_width="180"
month="{cp:variable:calendar_month}" year="{cp:variable:calendar_year}" cache="3600"}
```

Und in dem dazugehörigen Artikel mit der ID "17" ist dann folgender Container enthalten:

```
{cp:container module="[MODULNAME]" type="show_list_day" day="{cp:variable:calendar_day}"
month="{cp:variable:calendar_month}" year="{cp:variable:calendar_year}"}
```

Damit ist bereits ein voll funktionsfähiger Kalender vorhanden, der nun durch die Modifizierung der einzelnen Parameter den eigenen Wünschen gemäß angepasst werden kann.



In den hier gezeigten {cp:tags} ist jeweils statt dem internen Ordnernamen eines Modules "[MODULNAME]" enthalten. Da das Kalender-Modul auch mehrfach, d.h. mit verschiedenen Namen betrieben werden kann, müssen Sie hier "[MODULNAME]" noch durch den von Ihnen verwendeten Namen ersetzen.

5.12.4. Container: Tagesauflistung der Einträge generieren

Dieser Container vom Typ "show_list_day" dient der Anzeige der Einträge eines bestimmten Tages. Dazu werden dem

Container als einzige Parameter Tag ("day"), Monat ("month") und Jahr ("year") der Anzeige übergeben. Alles weitere (wie z.B. die Wahl der jeweils richtigen Vorlagen für die Darstellung der Einträge) wird intern durch den Container vorgenommen.

```
{cp:container module="[MODULNAME]" type="show_list_day" day="05" month="03" year="2005"}
// Anzeige der Einträge vom 05.03.2005
```

```
{cp:container module="[MODULNAME]" type="show_list_day" day="{cp:variable:calendar_day}"
month="{cp:variable:calendar_month}" year="{cp:variable:calendar_year}"}
// wie oben, jedoch werden hier die Parameter direkt der URL entnommen
```

Folgende Parameter sind für diesen Container zulässig / erforderlich:

Parameter für diesen Container		
day	Tag der Anzeige.	erforderlich
month	Monat der Anzeige.	erforderlich
type	Für diesen Container muss als Wert "show_list_day" verwendet werden.	erforderlich
year	Jahr der Anzeige.	erforderlich

5.12.5. Container: Kalender auf der Webseite anzeigen

Dieser Container vom Typ "show_month" dient zur Anzeige des Kalenders in einer Monatsansicht samt einer Navigation zum Wechsel zwischen den Monaten. Über eine Vielzahl von Parametern kann dabei die Gestaltung des Kalenders angepasst werden (z.B. können Text- und Hintergrundfarben, Rahmen usw. angegeben werden).

```
{cp:container module="[MODULNAME]" type="show_month" list_id="[ID]" month="{cp:variable:calendar_month}"
year="{cp:variable:calendar_year}"}
```

Die oben enthaltenen Parameter sind dabei zwingend erforderlich. Alle anderen Parameter sind optional und werden - falls nicht gesetzt - durch Standardwerte überschrieben.

Parameter für diesen Container		
border_type	Typ des Rahmens der Tabelle (Standardwert: "solid").	optional
border_width	Dicke des Rahmens der Tabelle (Standardwert: "1px").	optional
cell_padding	Textabstand in Pixel (Standardwert: "2").	optional
cell_spacing	Zellenabstand in Pixel (Standardwert: "1").	optional
color_background	Farbangabe für den Hintergrund der Tabelle (Standardwert: "transparent").	optional
color_border	Farbangabe für den Rahmen der Tabelle (Standardwert: "transparent").	optional
color_empty	Farbangabe für leere Felder (Standardwert: "transparent").	optional
color_font	Farbangabe für die Schrift (Standardwert: "transparent").	optional
color_font_title	Farbangabe für die Schrift im Titel (Standardwert: "#000000").	optional
color_head	Farbangabe für die Kopfzeile (Standardwert: "#000000").	optional
color_new_days	Farbangabe für noch folgende Tage (Standardwert: "transparent").	optional
color_old_days	Farbangabe für zurückliegende Tage (Standardwert: "transparent").	optional
color_this_day	Farbangabe für den aktuellen Tag (Standardwert: "transparent").	optional
current_day	Aktuellen Tag manuell festlegen.	optional
current_month	Aktuellen Monat manuell festlegen.	optional
current_year	Aktuelles Jahr manuell festlegen.	optional
list_id	ID des Artikels, in welchem die Einträge angezeigt werden.	erforderlich
month	Monat der Anzeige.	optional
table_width	Breite des gesamten Kalenders (Standardwert: "100%").	optional
type	Für diesen Container muss als Wert "show_month" verwendet werden.	erforderlich
year	Jahr der Anzeige.	optional

5.13. Links

*Das Link-Modul von **Contentpapst** kann zur Archivierung von Webseiten genutzt werden, welche von Ihren Besuchern in Form eines Kataloges abgerufen werden können. Jedoch sind ebenso Downloadarchive usw. mit dem System umsetzbar.*

5.13.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=links	Aufruf über den Administrationsbereich
index.php?file=links	Aufruf über den öffentlichen Bereich
links.html	Aufruf über den öffentlichen Bereich (Suchmaschinenfreundliche URL's)

5.13.2. "Top"-Markierung für die Links

Besonders herausragende Links können redaktionell mit einer "Top"-Markierung gekennzeichnet werden. Diese Markierung

kann nur im Administrationsbereich geändert werden, Besucher, welche Links einsenden, sind dazu nicht berechtigt. Für die "Top"-Markierung wird immer die Grafik "top.gif" im jeweils ausgewählten Vorlagen-Set genutzt. D.h. ist ein Vorlagen-Set mit dem Namen "cp" gewählt, so wird die Grafik unter "/themes/cp/image/top.gif" geladen.

5.13.3. Fehler melden

Ihre Besucher haben bei den Links die Möglichkeit, Ihnen Hinweise auf fehlerhafte Links mittels der Schaltfläche "Fehler melden" mitzuteilen. Diese Links werden zwar weiterhin angezeigt, jedoch finden Sie im Administrationsbereich eine entsprechende Meldung. Anschließend können Sie den Link manuell bzw. per automatischem Linkchecker überprüfen lassen.

5.13.4. Verfügbarkeit der Webseiten überprüfen

Neben der manuellen Meldung von Fehlern durch die Besucher Ihrer Webseite kann im Link-Modul in der Menüleiste unter "Links überprüfen" eine automatische Überprüfung der eingetragenen Webseiten vorgenommen werden. Hierzu ruft Contentpapst den Status der Webseiten in einem kurzen Intervall ab und zeigt diesen dann mit einer Farbmarkierung an.

Beim Aufruf der Funktion wird die Prüfung automatisch für 100 eingetragene Webseiten vorgenommen. Mittels "Auswahl begrenzen" können die Überprüfung und auch die Anzeige spezifiziert werden. Lassen Sie beispielsweise die kompletten Einträge durchsuchen und nur fehlerhafte Einträge ausgeben.

5.14. Medienbibliothek

Die Medienbibliothek gewährt Ihnen Zugriff auf die vom System genutzten Medien wie z.B. Grafiken oder Videos. Wie in der Dateiverwaltung navigieren Sie durch das Dateisystem und wählen die gewünschte Datei aus. Auf Linux/Unix-Systemen ist ein Wechsel der Zugriffsrechte einzelner Dateien möglich.

5.14.1. Allgemeines zu diesem Modul

Die Navigation innerhalb der Medienbibliothek ist ähnlich der Navigation der Dateiverwaltung. Sie bewegen sich innerhalb der Verzeichnisstrukturen Ihres Webspaces, ausgehend vom Contentpapst-Stammverzeichnis. Es werden in der Medienbibliothek nur Dateien mit Endungen angezeigt, welche auf einen Inhalt aus den Bereichen Bild, Video oder Sound schließen lassen.



Mit einem Klick auf das jeweils eingeblendete Bild gelangt man zur Detailansicht. Bei Flash-Filmen ist ein Klick auf den Textlink "Details" notwendig. In der Detailansicht sind Informationen über verschiedene Änderungs- und Erstellungsdaten der Datei, sowie deren Bild- und Dateigröße sichtbar.

Auf Linux-/Unix-Systemen können zusätzlich die Zugriffsrechte der Dateien geändert werden. Hierbei wird nach Rechten für Besitzer (owner), Gruppe (group) und Welt (all) unterschieden. Es existieren Lese- und Schreibrechte, sowie das Recht die Datei auszuführen. Weitere Informationen über den praktischen Einsatz von Unix-Dateirechten erhalten Sie von Ihrem Serveradministrator bzw. in Dokumentationen/Handbüchern zu Linux/Unix.

Aufgrund der unterschiedlichen Player für Video- und Audiodateien werden diese in der Medienbibliothek nicht direkt angezeigt. An ihrer Stelle taucht eine Grafi auf, welche auf den Inhalt der Dateien hinweist. Erst in der Detailansicht lassen sich Video- und Audiodateien mit einem Klick auf die erwähnte Grafik betrachten. Es öffnet sich hierzu ein neues Fenster, in welchem die Datei ausgeführt wird. Je nach der lokal auf dem Rechner des Clients installierten Software muss die Datei eventuell vorher heruntergeladen werden.

Unterstützte Dateitypen
AU, AVI, BMP, GIF, JPG, JPEG, MID, MOV, MP3, MPE, MPEG, MPG, OGG, PNG, QT, SWF, WAV

Wichtige URL's für dieses Modul	
admin.php?file=media	Aufruf über den Administrationsbereich

5.14.2. Suchen in der Medienbibliothek

Sollten Sie einmal den Überblick über den Datenbestand verloren haben, können Sie die leistungsfähige Suchfunktion der Medienbibliothek einsetzen. Diese durchsucht ausgehend vom Contentpapst-Stammverzeichnis die Ordner bis in eine festgelegte Tiefe (voreingestellt sind 2 Ebenen). Es kann zwischen Groß-/Kleinschreibung unterschieden werden.

Als Suchbegriff kann neben einfachen Wörtern auch ein regulärer Ausdruck genutzt werden. Unter "regulären Ausdrücken" versteht man Abfragen, welche besondere Steuerzeichen zur Einschränkung der Suche beinhalten. So können z.B. Wildcards eingesetzt werden oder das gesuchte Wort muss am Anfang oder Ende des Dateinamens stehen...

Anwendung der Suchfunktion	
*	Wildcard für beliebige Zeichen (z.B.: "c*s" trifft auch auf "cms" zu)
^	Suchbegriff am Anfang des Dateinamens (z.B.: "^cms" trifft zu auf "cms_backup.gif" usw.)
$	Suchbegriff am Ende des Dateinamens (z.B.: ".gif$" zeigt alle Dateien mit dieser Endung)
(n1|n2|n...)	Alle Möglichkeiten werden gematcht (z.B.: "cat(01|02)" trifft zu auf "cat01" und "cat02")
^...$	Jede Datei mit genau "." Zeichen (hier z.B. drei Zeichen)
li.t	"." steht für ein beliebiges Zeichen (trifft hier z.B. für "list" zu)

5.15. Navigation

Die Navigation verbindet die einzelnen Bereiche Ihrer Webseite zu einem Ganzen und erlaubt dem Besucher einen schnellen Wechsel zwischen den verschiedenen Seiten. Dadurch erhält die Navigation bei der Erstellung einer Webseite eine besondere Bedeutung.

5.15.1. Allgemeines zu diesem Modul

Das Modul zur Erstellung einer Navigation für Ihre Webseite erreichen Sie über den Button "Navigation" auf der linken Seite des Administrationsbereiches. Dieses Modul ist zweigeteilt in eine Übersicht der bereits vorhandenen Navigationspunkte und verschiedene Menüs zum Anlegen, Bearbeiten und Löschen dieser Navigationspunkte.

In der Übersicht sehen Sie anhand einer Baumstruktur, wie die einzelnen Navigationspunkte miteinander verknüpft sind und in welcher Reihenfolge diese angezeigt werden. Für jeden Navigationspunkt gibt es je nach Position die Möglichkeit, diesen nach oben oder unten zu verschieben. Auch ein direktes Löschen ist möglich. Sollten Sie versuchen einen Navigationspunkt zu löschen, welcher noch über Unterpunkte verfügt, dann ist eine Bestätigung des Löschvorganges notwendig. Alle verknüpften Navigationspunkte werden anschließend automatisch gelöscht.

Um einen Navigationspunkt zu erstellen bzw. zu bearbeiten, müssen Sie zuerst einen Titel und das Ziel (die URL) eingeben, einen darüberliegenden Navigationspunkt auswählen (sofern dieser nicht in der ersten Ebene liegt) sowie den Status festlegen. Um direkt auf einen Artikel oder eine News zu verlinken, können Sie alternativ auch die Schnellauswahl nutzen.

Die Möglichkeiten zur Darstellung der Navigation (bzw. einzelner Teile des Navigationsbaumes) sind äußerst vielfältig. Auch die Erstellung komplexer Navigationen auf der Basis von Javascript, Flash oder DHTML ist mit einiger Übung kein Problem. In unserer Knowledge Base (http://www.sandoba.de/support/knowledge-base/) haben wir einige Artikel zu diesem Thema für Sie zusammengestellt.

Wichtige URL's für dieses Modul	
admin.php?file=navigation	Aufruf über den Administrationsbereich

5.15.2. Container: Navigation auf der Webseite anzeigen

Die Navigation ist das Herzstück einer jeden Webseite. Sie kann äußerst unterschiedliche Formen annehmen. Doch über einen Container lassen sich in Contentpapst fast beliebige Navigationen mit einer gewünschten Tiefe realisieren. Ausführliche Anleitungen zu verschiedenen Navigationstypen finden sich auch in der Knowledge Base und im Entwickler-Blog.

Der Containeraufruf "{cp:container module="navigation"}" ermöglicht Angaben zur Kategorie, den zu verwendenden Vorlagen, dem Stil usw.:

Parameter für diesen Container		
block	Liste der nicht sichtbaren Navigations-Punkte (kommasepariert)	optional
cache	Cache-Zeit der Navigation in Sekunden	optional
style	Stil (nur wenn Standardnavigation genutzt wird)	optional
category	Übergeordnete Kategorie (Oberste Ebene ist "0", komma-separierte Angabe für mehrsprachige Seiten, z.B. "1,12")	optional
number	alternativ zu *category* für die direkte Anzeige eines Unterpunktes	optional
order	Links ordnen (möglich sind "date", "hits", "number" und "title")	optional
type	Verwendete Tabelle ("news", "article", "navigation" oder leer)	optional
template	Pfad zu einer Vorlage	erforderlich
template_selected	Pfad zur Vorlage, wenn ein Navigationspunkt ausgewählt wurde, *1	optional
template_subnodes	Pfad zur Vorlage, wenn Unterpunkte vorhanden sind	optional

*1 Für "template_selected" muss zusätzlich über den Parameter "selected_field" das Feld der Datenbanktabelle angegeben werden, welches die URL enthält. Bei der Navigationstabelle muss dies also "selected_field="nav_url"" sein.

5.15.3. Eigene Navigationen anlegen

Neben den Standardnavigationen, die vor allem dem einfachen Einstieg in die Thematik der {cp:tags} gedacht sind, bietet Contentpapst fast unbegrenzte Möglichkeiten zur Gestaltung individueller Navigationen. Dies ist besonders wichtig, denn die Navigationen sind es, die dem Benutzer die Struktur einer Webseite näherbringen und diesen anleiten, weitere Bereiche zu erkunden.

Wichtig für das Verständnis der Funktionsweise der {cp:tags} ist das Prinzip der Rekursivität. Über die {cp:tags} können Inhalte wie z.B. ein Navigationspunkt aufgerufen werden, die über eine Vorlage gestaltet werden. In dieser Vorlage können wiederum {cp:tags} enthalten sein, die eine eigene Vorlage aufrufen usw. Dies kann beliebig fortgeführt werden, bis in einer Vorlage keine {cp:tags} mehr vorhanden sind bzw. das Ergebnis leer ist. In einer Vorlage können beliebig viele {cp:tags} enthalten sein.

Wir werden uns dieses Prinzip der Rekusivität beim Aufbau einer Navigation zu nutze machen. Nach Vollendung werden Sie über eine Navigation verfügen, die frei an die eigenen Bedürfnisse angepasst werden kann. Mit dem Wissen über den Aufbau von Navigationen können Sie dann auch komplett eigene Navigationen konzipieren und verwirklichen.

Zuerst muss festgestellt werden, dass in der Navigation jeder Navigationspunkt eine eindeutige ID besitzt. Im Navigations-Modul von Contentpapst ist diese ID direkt in der Übersicht sichtbar. Wenn Sie eine Navigation auf Ihrer Webseite aufbauen wollen, müssen Sie jetzt wählen, welcher Navigationspunkt der Ursprung Ihrer Navigation sein soll. Der allgemeine Ursprung hat die ID "0".

Im nächsten Schritt wählen Sie die Vorlage aus, in welchem die Navigation platziert werden soll. Generell ist dies die Vorlage "main.htm" aus Ihrem verwendeten Vorlagen-Set. Dort wird ein {cp:tag} platziert, der Contentpapst über verschiedene Parameter Anweisungen gibt, welche Navigationspunkte angezeigt werden sollen. Welche Parameter dies genau sind, erfahren Sie im Punkt "{cp:container="navigation"}" (siehe Anleitung für {cp:container}).

Beginnen wir mit der konkreten Umsetzung, indem dieser {cp:tag} in die Vorlage eingebaut wird:

```
{cp:container module="navigation" category="0" template="hauptkategorien.htm"
template_selected="aktuellerpunkt.htm" selected_field="nav_url"}
```

Der Tag bewirkt folgendes: Contentpapst ruft intern die Navigationspunkte auf, die in der ersten Ebene unterhalb der Kategorie/ID "0" liegen, d.h. die Hauptpunkte. Diese werden über die Vorlage "hauptkategorien.htm" gestaltet oder über die Vorlage "aktuellerpunkt.htm", wenn die URL des Navigationspunktes mit der aktuellen URL übereinstimmt (also wenn der Navigationspunkt ausgewählt ist). Die Vorlagen bauen sich wie folgt auf:

Vorlage "hauptkategorien.htm":

```
<div class="navigation_hauptkategorie">» <a href="{cp:navigation:nav_url}" title="{cp:navigation:nav_title}">
{cp:navigation:nav_title}</a></div> {cp:container module="navigation" category="{cp:navigation:nav_id}"
template="unterkategorien.htm"}
```

Vorlage "aktuellerpunkt.htm":

```
<div class="navigation_aktuell">{cp:navigation:nav_title}</div> {cp:container module="navigation"
category="{cp:navigation:nav_id}" template="unterkategorien.htm"}
```

Über die Tags {cp:navigation:nav_id}, {cp:navigation:nav_title}, {cp:navigation:nav_url}, {cp:navigation:nav_base}, {cp:navigation:nav_status} und {cp:navigation:nav_pos} werden die einzelnen Inhalte eines Navigationspunktes wie z.B. der Titel und die ID eingefügt. Die ID ("{cp:navigation:nav_id}") dient in beiden Vorlagen gleichzeitig dazu, die nächste Ebene der Navigationspunkte aufzurufen, die direkt darunter liegen. Für diese Punkte wird die Vorlage "unterkategorien.htm" verwendet:

Vorlage "unterkategorien.htm":

```
<div class="navigation_unterkategorie"><a href="{cp:navigation:nav_url}" title="{cp:navigation:nav_title}">
{cp:navigation:nav_title}</a></div>
```

Jetzt haben Sie bereits eine fertige Navigation, die sich selbstständig aktualisiert und mittels HTML und CSS gestaltet ist. Die notwendigen CSS-Regeln sind folgende:

```
.navigation_hauptkategorie {
    width: 180px;
    background: #728E92;
    padding: 3px;
    font-weight: bold;
    font-family: Tahoma;
    color: #FFFFFF;
    border-bottom: 1px solid #FFFFFF;
}

.navigation_unterkategorie a {
    width: 180px;
    background: #F5F5F5;
    padding: 3px;
    padding-left: 14px;
    font-family: Tahoma;
    color: #000000;
    border-bottom: 1px solid #FFFFFF;
}

.navigation_aktuell {
    width: 180px;
    background: #557479;
    padding: 3px;
    padding-left: 14px;
    font-weight: bold;
    font-family: Tahoma;
    color: #FFFFFF;
    border-bottom: 1px solid #FFFFFF;
}

.navigation_unterkategorie a:hover {
    color: #000000;
}

.navigation_hauptkategorie a:hover, .navigation_aktuell a:hover {
    text-decoration: none;
}

.navigation_hauptkategorie a:link, .navigation_hauptkategorie a:hover, .navigation_hauptkategorie a:visited {
    color: #FFFFFF;
}

.navigation_unterkategorie a:hover {
    background: #F1F1F1;
    text-decoration: none;
}
```

Auf der linken Seite sehen Sie ein Beispiel der fertigen Navigation mit den Navigationspunkten einer Webseite zum Thema "Reiser & Abenteuer", die mittels der obenstehenden CSS-Regeln gestaltet wurde. Die einzelnen Haupt- und Unterpunkte können klar voneinander unterschieden werden, ebenso ist die aktuelle Seite anders eingefärbt. Die Navigationspunkte besitzen zusätzlich

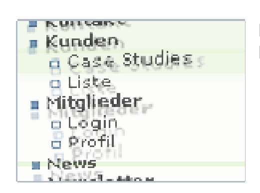


einen Mouseover-Effekt.

Jetzt wissen Sie, wie Sie eine Navigation mit Contentpapst aufbauen. Dieses Beispiel kann beliebig angepasst und leicht z.B. um eine dritte und vierte Ebene oder weitere Vorlagen ergänzt werden. Prinzipiell kann alles realisiert werden, was mittels CSS, (X) HTML, DHTML, Javascript und Co. möglich ist. Sie können sogar XML erzeugen und dies als Grundlage für eine Flash-Navigation verwenden.

Noch ein Tipp: Um die Ladezeit für das Navigationsmenü erheblich zu senken, sollten Sie in der Hauptvorlage (z.B. der "main.htm") den zusätzlichen Parameter "cache" integrieren. Geben Sie für diesen z.B. "86400" als Wert an, dann wird die gesamte Navigation für einen Tag zwischengespeichert.

5.15.4. Standardmenü: Baummenü



Das Baummenü ist ein raffiniertes Menü, welches sich besonders zur Darstellung einer Sitemap eignet. Auf Wunsch kann das Menü auf- bzw. eingeklappt dargestellt werden.

```
{cp:container module="navigation" style="standard"} - Eingeklapptes Menü
{cp:container module="navigation" style="standard-full"} - Aufgeklapptes Menü
```

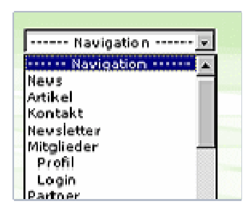
5.15.5. Standardmenü: Buttons



Hier werden dynamische Grafiken für den Aufbau der Navigation genutzt. Angepasst werden kann dieser Navigationstyp durch eine eigene Hintergrund-Grafik. Tauschen Sie hierfür die Datei "nav_back.jpg" im Ordner "images" des genutzten Vorlagen-Sets aus. Für den Einbau in die Vorlagen stehen Ihnen die folgenden {cp:tags} zur Verfügung:

```
{cp:container module="navigation" style="buttons"} - Horizontales Menü
{cp:container module="navigation" style="buttons-ver"} - Vertikales Menü
```

5.15.6. Standardmenü: Dropdown-Menü



Das Dropdown-Menü kann beispielsweise am Ende einer Seite eingesetzt werden. Dadurch hat der Besucher die Möglichkeit, die Seite zu wechseln, ohne zum Beginn zurückkehren zu müssen. Für das Dropdown-Menü gibt es zwei mögliche {cp:tags}:

```
{cp:container module="navigation" style="list"} - Navigation nur mit Hauptmenüpunkten
{cp:container module="navigation" style="list-full"} - Navigation mit allen Menüpunkten
```

5.15.7. Standardmenü: Links



Diese Navigationsart listet die Menüpunkte nebeneinander bzw. untereinander als Textlink auf. Es werden nur die Hauptmenüpunkte gezeigt. Auch hier gibt es wieder zwei mögliche {cp:tags}:

```
{cp:container module="navigation" style="simple"} - Horizontales Menü
{cp:container module="navigation" style="simple-ver"} - Vertikales Menü
```

5.15.8. Zusätzliche Navigationstypen

Contentpapst bietet bereits einige vorgefertigte Navigationstypen an, welche auf der Hauptseite der Navigationseinstellungen auch näher beschrieben werden. Die Navigationstypen bauen im Wesentlichen auf der durch {cp:container} übermittelten Variable "type" auf. Es wird eine Liste generiert, dessen Gestaltung durch jeweils eigene Vorlagen übernommen wird.

Neue Navigationstypen können über {cp:container="navigation"} angelegt werden (siehe Anleitung für {cp:container}).

5.16. News-System

Das News-System bietet alle Funktionen, um einen News-Ticker mit kurzen Meldungen aufzubauen. Erweiterte Funktionen wie eine Laufzeit-Beschränkung und Kommentare stehen ebenfalls zur Verfügung.

5.16.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=news	Aufruf über den Administrationsbereich
index.php?file=news	Aufruf über den öffentlichen Bereich
news.html	Aufruf über den öffentlichen Bereich (Suchmaschinenfreundliche URL's)

5.16.2. Kommentare

Kommentare von Besuchern werden je nach getroffener Einstellung ("News > Modul-Einstellungen") sofort online gestellt bzw. müssen vorher von einem Redakteur freigeschalten werden. Die Möglichkeit Kommentare zu schreiben kann mittels der Vorlage für die News-Detailansicht eingeschränkt oder ganz unterbunden werden.

Neu eingetroffene und noch nicht bestätigte Kommentare werden im News-Modul im unteren Bereich zusammen mit neuen Newseinträgen unter "Meldung" angezeigt. Zu den Kommentaren einer News gelangt man, indem man diese über "News bearbeiten" auswählt. Kommentare können bearbeitet, gelöscht und im Status (geschützt/öffentlich verfügbar) geändert werden.

5.16.3. Laufzeit eingrenzen

Die Angabe der Laufzeit einer News im Format "TT.MM.JJ" (z.B. 20.03.2003) schränkt die Sichtbarkeit dieser News im öffentlichen Bereich ein. Es kann ein Anfangs- und ein Enddatum bzw. beides eingegeben werden. So können News verfasst und automatisch z.B. erst eine Woche später veröffentlicht werden.

Ebenso lässt sich eine zeitlich nur begrenzt gültige News (z.B. News über die Teilnahme an einem Gewinnspiel) nach Ablauf einer Zeit von der Webseite entfernen. Es wird immer der Beginn des jeweils angegebenen Tages (z.B. 20.03.2003 0:00) für die Anzeige der News genutzt.

5.16.4. News hinzufügen

Um News in das System zu integrieren, greifen Sie auf das News-System zurück. Der Bereich "News hinzufügen" bietet Ihnen hier alle benötigten Optionen, um eine News nach den eigenen Wünschen in das System einzustellen. News können Kategorien zugeordnet, besonders gekennzeichnet, mit einem Bild versehen oder auch in der Laufzeit beschränkt werden. Die zuletzt genannte Funktion erlaubt es, eine News nur für einen bestimmten Zeitraum auf der Webseite für den Besucher sichtbar anzuzeigen.

5.16.5. News in den letzen "n" Tagen

Möchte man dem Besucher gezielt Übersichten über News in den letzten "n" Tagen präsentieren (z.B. im letzten Monat), so muss an "index.php?file=news" die Variable "days" mit einer Zahlenangabe angehängt werden. Diese URL wird dann an der gewünschten Stelle in der Navigation bzw. in den Vorlagen genutzt.

"index.php?file=news&days=7"	Beispiel für News der letzten Woche
"index.php?file=news&days=31"	Beispiel für News im letzten Monat

5.16.6. Zugriffssperre

Um sicherzustellen, dass Arbeiten an den News nicht behindert werden, wird der Zugriff auf die News in der Datenbank gespeichert. Greift ein zweiter Redakteur auf eine bereits durch einen anderen Redakteur geöffnete News zu, so wird diesem der Zugriff verwehrt. Ebenso wird bei der Löschung von News verfahren. Der Zugriff auf die News wird entweder durch die Speicherung wieder freigegeben oder durch den Ablauf einer festgelegten Zeitspanne.

Bei Contentpapst sind dies standardmäßig 30 min. D.h. wenn nach dieser Zeit noch keine Speicherung erfolgt ist, wird die News dennoch wieder zur Bearbeitung für andere Redakteure freigegeben. Durch diese Regelung kann die Redaktionsarbeit ohne Behinderungen und bei gleichzeitiger Sicherung der News erfolgen. Sollte eine schnellere Freigabe nötig sein, haben Sie immer die Möglichkeit, mit dem zuständigen Redakteur Kontakt aufzunehmen.

5.17. Newsletter

*Der Newsletter-Bereich von **Contentpapst** dient dem Versenden von Informationen über Ihre Webseite in regelmäßigen oder unregelmäßigen Abständen. Es kann frei gewählt werden, ob man über einen Newsletter einen geschlossenen Kundenkreis informiert oder eine öffentlich zugängliche Liste betreibt.*

5.17.1. Allgemeines zu diesem Modul

Im Newsletter-Modul besteht eine Trennung zwischen Newsletter und Newsletter-Ausgabe. Um Newsletter zu verschicken, legt man zuerst einen neuen Newsletter mit Titel, Beschreibung, Kopf- und Fußbereich sowie einem optionalen Text für Begrüßungsmail und Abschiedsmail an. Auf der Hauptseite des Newsletter-Modules kann dann eine Newsletter-Ausgabe zusammengestellt werden.

Der Newsletter, dessen Kopf- und Fussbereich verwendet werden soll, muss hier zuerst ausgewählt werden. Der Newsletter wird an alle für diesen Newsletter eingetragenen Abonnenten verschickt. Nach dem Versand des Newsletters wird dieser in das entsprechende Archiv eingefügt.

Wichtige URL's für dieses Modul	
admin.php?file=newsletter	Aufruf über den Administrationsbereich
index.php?file=newsletter	Aufruf über den öffentlichen Bereich
newsletter.html	Aufruf über den öffentlichen Bereich (Suchmaschinenfreundliche URL's)

5.17.2. Autom. Zeilenumbruch

Ist die Funktion "Automatischer Zeilenumbruch" bei der Erstellung einer Newsletter-Ausgabe aktiviert, wird der eingegebene Text im Newsletter beim Versenden automatisch umgebrochen. D.h. Zeilen weisen in der finalen Fassung maximal eine Länge von 75 Zeichen auf und sind somit in jedem Email-Programm ohne horizontales Scrollen lesbar. Diese Funktion wirkt sich nur auf die Text-Version des Newsletters aus. In einer HTML-Version hätte die Funktion u.U. Darstellungsprobleme zu Folge, da eventuell HTML-Codes getrennt werden.

5.17.3. Begrüßungs- und Abschiedsmail

Um den Newsletter vor unerwünschten Anmeldungen zu schützen, kann es notwendig sein, eine Begrüßungsmail an den neu angemeldeten Abonnenten zu verschicken. Diese Email enthält eine automatisch generierte URL, welche der Abonnent anklicken muss, um freigeschalten zu werden. Entsprechend ist der Ablauf bei einer Abmeldung vom Newsletter durch den Abonnenten. Gleichzeitig jedoch dienen diese Emails auch zu einer besseren Bindung der Abonnenten zum Newsletter. Dies kann beispielsweise durch eine persönliche Ansprache noch erhöht werden.

5.17.4. Benachrichtigungsfunktion

Sie können das System anweisen, Sie per Email zu benachrichtigen, sobald ein Besucher den Newsletter bestellt. Ebenso kann Sie das System bei Abmeldungen benachrichtigen. So bleiben Sie immer über das aktuelle Abonnentenverhalten informiert.

5.17.5. Datei(en) anhängen

Sie können jeder Ausgabe eines Newsletters eine Datei anhängen, beispielsweise ein Bild zu einer Pressemeldung oder auch Produktdetails in Form einer PDF-Datei. Bitte beachten Sie, dass der Anhang des Newsletters nur von Abonnenten gelesen werden kann, deren Email-Account entsprechende Optionen unterstützt.

5.17.6. Email an das Mitglied schicken

Aus dem Administrationsbereich heraus können Sie eine Email an einzelne Newsletter-Abonnenten schicken. Angegeben werden können hier neben dem Titel und dem Text der Email auch Email-Adressen für den Versand der Email als Kopie- bzw. Blind-Kopie. Nutzen Sie diese Funktion bei Ihren Abonnenten, um einen individuellen Kontakt zu pflegen.

5.17.7. Filter (E-Mail / Domains sperren)

Sie können über den Filter einzelne Email-Adressen oder auch komplette Domainnamen von der Anmeldung zum Newsletter ausschließen. Für ganze Domainnamen geben Sie bitte "@domainname.de" ein. Auch einzelne Wortbestandteile können gefiltert werden. Wird bei einer Anmeldung ein derartiger Bestandteil entdeckt, wird diese Anmeldung abgewiesen.

5.17.8. HTML- oder Text-Newsletter

Der Newsletter kann komplett im HTML-Format versendet werden oder - wie oftmals bevorzugt - im reinen Text-Format. Je nach Ausrichtung Ihrer Webseite können Sie so Ihre Abonnenten optimal mit den passenden Inhalten versorgen. Um einen Newsletter im HTML-Format zu versenden, wählen Sie im Newsletter-Bereich als Versandart "HTML" aus. Die notwendigen Änderungen im Quelltext werden automatisch beim Versand vorgenommen.

5.17.9. Import von Mitgliedsdaten

Sollten Sie bereits über einen bestehenden Datenstand an Email-Adressen aus einem Newsletter etc. verfügen, so können Sie diesen mittels der Funktion "Mitgliedsdaten importieren" in Contentpapst einbinden. Gehen Sie hierzu im Newsletter-Bereich auf die Mitgliedsverwaltung.



Hier haben Sie die Auswahl zwischen TXT- und CSV-Dateien. Letztere können Sie direkt aus Microsoft Outlook oder Excel übernehmen. Beachten Sie dabei, dass Sie nur Vorname, Name und Email-Adresse eines Eintrages exportieren, sonst entstehen Fehler beim Import der Daten. Zusätzlich zur Quelldatei haben Sie die Möglichkeit, Einstellungen bezüglich des Newsletters und des Mitglieder-Status zu treffen. Durch die Eingabe eines Trennzeichens können Sie auch TXT-Dateien einlesen, in denen die Email-Adressen durch andere Zeichen getrennt werden.

5.17.10. Kopf- und Fuß-Bereich

Für jeden Newsletter können separate Kopf- und Fuß-Bereiche angelegt werden, welche in den einzelnen Ausgaben dann jeweils am Anfang bzw. Ende automatisch eingefügt werden. Diese Bereiche des Newsletters können durch untenstehende Elemente angepasst werden. U.a. ist es möglich, den Newsletter durch eine persönliche Anrede zu personalisieren oder Daten wie die Anzahl der Abonnenten einfügen zu lassen.

{cp:tags} für den Kopf- und Fuß-Bereich des Newsletters	
{cp:title}	Titel der Webseite

{cp:date}	Datum einsetzen
{cp:user}	Anzahl der Abonnenten
{cp:issue}	Nummer der Ausgabe
{cp:personal}	Persönliche Anrede
{cp:email}	Email-Adresse des Abonnenten
{cp:url}	URL der Webseite
{cp:unregister}	Abbestellen des Newsletters
{cp:archive}	Newsletter-Archiv

5.17.11. Lesebestätigung

Wenn Sie genau wissen wollen, wie wichtige Newsletter bei Ihren Lesern ankommen, d.h. ob diese gelesen werden, können Sie zusätzlich die Lesebestätigung einschalten. Unter Email-Programmen, welche diese Funktion unterstützen, wird der Leser beim öffnen der Email aufgefordert, eine Lesebestätigung abzusenden. Je nach Einstellung des Programmes kann diese auch automatisch gesendet werden. Bedenken Sie jedoch, dass die Lesebestätigung nicht immer automatisch gesendet wird und Sie somit nicht von allen Abonnenten eine Email zurückerhalten werden.

5.17.12. Mitglieder einladen

Wenn Sie über Email-Adressen von Personen verfügen, welche noch nicht Ihren Newsletter abonniert haben, können Sie diese über die Mitgliederverwaltung direkt anschreiben. Neben der Angabe von Titel (Betreff) und Text der Email schreiben Sie die Email-Adressen getrennt durch Komma auf. An jede dieser Email-Adressen wird dann eine Email versendet.

5.17.13. Mitgliederverwaltung

Die Abonnenten der Newsletter werden ähnlich wie in der eigentlichen Mitgliederverwaltung des Systemes gelistet. Diese Trennung ist notwendig, da oftmals Abonnenten des Newsletters nicht mit den angemeldeten Mitgliedern gleichzusetzen sind. Jedoch können angemeldete Mitglieder in Ihrem Profil direkt Newsletter bestellen oder kündigen.

In der Mitgliederverwaltung sind jeweils 50 Mitglieder pro Seite aufgelistet. Diese können nach Name, Email-Adresse, Anmeldungsdatum, dem bestellten Newsletter und ihrem jeweiligen Status sortiert werden. Neben einer Suchfunktion lassen sich in diesem Bereich Mitgliedsdaten importieren sowie Emails direkt an einzelne Mitglieder schicken. Weiterhin lässt sich jedes Mitglied bearbeiten und aus der Liste wieder löschen.

5.17.14. Prioritäten

Newsletter-Ausgaben können zusammen mit einer Priorität versendet werden. Hierzu wird dem Newsletter eine zusätzliche Header-Angabe "X-Priority" hinzugefügt. Allerdings ist die Angabe von Prioritäten in Emails nicht standardisiert, so dass nicht alle Email-Programme diese auch verarbeiten können. Im oft genutzten Outlook (z.B. Outlook Express) von Microsoft (TM) sind drei Prioritäten möglich: Hoch, Normal und Niedrig.

5.17.15. Schutz gegen Spamming

Um einer unerwünschten Flut von fehlerhaften Anmeldungen vorzubeugen, überprüft das System bei der Anmeldung die übermittelte Email-Adresse auf ihre Korrektheit. Es wird u.a. geprüft, ob eine Domain mit einer gültigen Domainendung sowie das "@"-Zeichen in der Email-Adresse vorhanden sind. Doppelten Einträgen wird ebenfalls vorgebeugt.

Durch die Begrüßungsmail bei der Anmeldung muss der Newsletter-Account erst freigeschalten werden. Davor werden keine Newsletter an diese Email-Adresse versendet. Benutzer haben bis zu 7 Tage nach der Bestellung des Newsletters die Möglichkeit, ihr Abonnement zu bestätigen. Danach verfällt das Abonnement. Im Admin-Bereich können zudem gezielt Email-Adressen und Domainnamen gesperrt und einzelne Accounts gelöscht werden.

5.17.16. Tracking

Auf Wunsch kann die Beliebtheit der einzelnen im Newsletter enthaltenen Links per Tracking-Statistik ermittelt werden. Ist die Tracking-Funktion beim Versenden einer Newsletter-Ausgabe aktiviert, so werden alle im Newsletter vorhandenen Links durch eine dynamische Tracking-URL ersetzt. Klickt ein Abonnent des Newsletters einen solchen Link an, so wird er über die Tracking-Seite weiter auf die Original-Seite geleitet. Der Zugriff auf die Tracking-Seite wird dabei gespeichert und statistisch aufbereitet. Unter "Tracking" im Newsletter-Modul kann man Einsicht auf die Zugriffe erhalten.

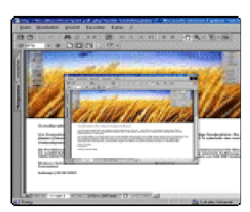
5.18. PDF-Export

Für den Einsatz einzelner Inhalte Ihrer Webseite im Print-Bereich benötigen Sie oftmals Dateien im PDF-Format, wie von Adobe spezifiziert. Diese können Sie hier "on-the-fly" generieren lassen und mit den gewünschten Angaben wie Titel, Thema, Schlüsselwörtern und auch einem Passwort versehen. Die generierten PDF-Dateien lassen sich z.B. problemlos bei Print-Publikationen einsetzen.

5.18.1. Allgemeines zu diesem Modul

Mittels des PDF-Exports lassen sich einzelne Artikel und News aus Contentpapst heraus als PDF-Datei erzeugen. Im öffentlichen Bereich kann hierfür der Tag {cp:pdf} verwendet werden. Ein Redakteur kann jedoch die Erstellung der PDF-Datei stärker beeinflussen. So können ein Titel, ein Thema und Schlüsselwörter integriert werden, welche beispielsweise bei der Platzierung in Suchmaschinen hilfreich sein können.

Auf Wunsch lässt sich die PDF-Datei mit einem Passwort schützen (40Bit RC4-Schlüssel). Beim öffnen der PDF-Datei wird dann



immer zur Eingabe des Passworts aufgefordert.

Wichtige URL's für dieses Modul	
admin.php?file=pdf	Aufruf über den Administrationsbereich

5.18.2. Auswahl Kategorie / Eintrag

Sie können News oder Artikel einzeln in eine PDF-Datei integrieren. Jedoch lassen sich auch komplette Kategorien oder mehrere News / Artikel aus verschiedenen Kategorien in eine Datei schreiben. Wählen Sie zuerst aus, ob Sie News oder Artikel in eine PDF-Datei integrieren wollen und anschließend selektieren Sie die gewünschten Kategorien / Einträge oder treffen Sie eine gemischte Auswahl.

5.18.3. Einstellungen

Generell gültige Einstellungen lassen sich getrennt von der Erstellung der PDF-Dateien im unteren Bereich des Modules vornehmen. Hier ist die Angabe eines Autors möglich, Links sowie Bilder können freigeschalten und eine Schriftart ausgewählt werden. Das Einfügen neuer Schrift ist momentan noch nicht möglich.

5.19. Sicherung

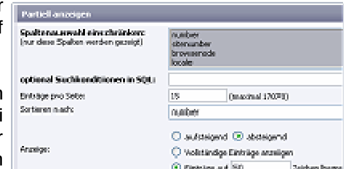
Gerade bei größeren Portalen oder Webseiten mit sensiblen Daten ist es wichtig, regelmäßige Backups der Datenbank und der Dateien zu machen. ***Contentpapst*** *liefert die passenden Funktionen gleich dazu. Updates für das System werden ebenfalls hier aufgespielt.*

5.19.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=backup	Aufruf über den Administrationsbereich

5.19.2. Anzeigen / Partiell Anzeigen

Einblick in die Struktur der Datenbanktabellen erhalten Sie, indem Sie in der Tabellenauflistung die gewünschte Datenbanktabellen auswählen und dort auf "Anzeigen" bzw. "Partiell Anzeigen" klicken.



"Anzeigen" öffnet sofort ein neues Fenster, in welchem die Datenbanktabelle in Spalten und Zeilen aufgeteilt dargestellt wird. Die Anzahl der Spalten ist dabei jeweils von der Struktur der Datenbanktabelle abhängig und kann unter Umständen seitliches Scrollen erfordern. Unterhalb der ersten Zeile werden dann alle enthaltenen Datensätze aufgelistet.

Standardmäßig ist die Auflistung auf 10 Einträge pro Seite begrenzt, Sie können jedoch mit der bekannten Navigation auf weitere Seiten umschalten. Die Datensätze lassen sich auf Wunsch auch ordnen. Hierzu wählen Sie einen der fettgedruckten Spaltennamen an. Durch einen Klick auf "Zurück" schließen Sie das Fenster wieder.

"Partiell Anzeigen" erweitert die Abfragemöglichkeiten. Hier können Sie zunächst die Anzahl der Datensätze pro Seite sowie die Länge der Datensätze eingrenzen und bereits eine Sortierung nach einem Spaltennamen vornehmen. Mit einer Auswahl im Feld "Spaltenauswahl einschränken" kann die Auswahl auf bestimmte Spalten begrenzt werden.

Diese Funktionen erlauben dem erfahrenen Nutzer eine bessere Übersicht über die in der Datenbank enthaltenen Datensätze. Es lässt sich durch diese Dastellung auch eine bessere Übersicht über die Verknüpfung der Tabellen untereinander und die Funktionsweise von Contentpapst erlangen.

5.19.3. Backup erzeugen

Die wichtigste Funktion des Sicherungs-Moduls ist natürlich die Backup-Funktion. Mit dieser lässt sich gezielt eine lokale Sicherung der in der Datenbank gespeicherten Datensätze durchführen.

Aufgelistet finden Sie im Modul die von Contentpapst genutzten Datenbanktabellen innerhalb der eingestellten Datenbank. Zu jeder Tabelle ist neben einigen (hier gesondert beschriebenen) Funktionen und der Anzahl der jeweiligen Datensätze eine Checkbox enthalten, mittels derer eine Tabelle ausgewählt werden kann. Ein Klick auf den Tabellennamen hat den gleichen Effekt. Alternativ kann man bei einem Backup der gesamten Datenbank auch alle Tabellen auf einmal markieren.

Wählen Sie nach der Auswahl der Tabellen die gewünschte Aktion aus. Möglich sind hierbei ein Backup der Datensätze (Daten), der Tabellen (Struktur) sowie ein komplettes Backup (Struktur/Daten). Sobald diese Wahl getroffen wurde, öffnet sich ein Fenster, welches zum Download der generierten Datei auffordert. Bei einer großen Datenmenge kann dieser Vorgang durchaus einige Minuten in Anspruch nehmen.

5.19.4. CSV-Datei exportieren

Die Inhalte von Datenbanktabellen können zur weiteren Verwendung in Excel und anderen Programmen in Form einer CSV-Datei exportiert werden. Hierzu wählen Sie die gewünschte Tabelle aus und gehen im Menü unter "Aktion auswählen" auf "Backup als CSV-Datei (einzeln)". Sie erhalten wie gewohnt die Aufforderung, die generierte Datei herunterzuladen.

Die Datensätze der Tabelle werden bei der Konvertierung automatisch so angeordnet, dass diese wie in der Online-Anzeige in Spalten/Zeilen getrennt sichtbar sind.

5.19.5. Datenbanken bearbeiten

Die Funktionen für die Verwaltung von Datenbanken sollten nur für den Administrator zugänglich sein, da hier bei unsachgemäßer Nutzung schnell ein Datenverlust auftreten kann. Diese Funktionen umfassen die Erstellung und Löschung von Datenbanken.

U.a. kann die Erstellung neuer Datenbanken sinnvoll sein, wenn zusätzliche Scripte/Systeme installiert werden sollen, jedoch nicht die aktuelle Datenbank dafür genutzt werden soll. Durch eine Löschung können Sie nicht mehr genutzte Datenbanken entfernen. Beachten Sie bitte, dass diese Funktionen von den Servereinstellungen abhängig sind und nicht jeder Hoster die Erstellung neuer / Bearbeitung vorhandener Datenbanken erlaubt.

5.19.6. Datenbanktabellen löschen / leeren

Zur Vorbereitung der Aufspielung eines Backups können Tabellen geleert werden. Hierbei werden alle enthaltenen Datensätze entfernt, die Tabellenstruktur selbst bleibt jedoch erhalten. Wenn größere Änderungen erfolgen sollen, können Tabellen zusätzlich komplett gelöscht werden. Beide Funktionen können direkt in der Tabellenauflistung angewählt werden.

5.19.7. Datenbanktabellen optimieren

Gerade nach umfangreichen Updates, d.h. wenn viele Datensätze eingefügt bzw. gelöscht wurden, ist es anzuraten, die betroffenen Tabellen zu optimieren. Dies bewirkt, dass entstandene Datenlücken geschlossen werden und ein Überhang von Daten abgebaut wird. Die Datenbank erlaubt nach einer Optimierung wieder schnellere Abfragen.

5.19.8. Größe der Datenbank / Tabellen

In der Auflistung ist für jede Tabelle neben der Anzahl der Einträge die jeweilige Größe angegeben. Diese errechnet sich aus dem von Struktur und Daten belegten Speicherplatz. Ganz am Ende der Auflistung ist die Größe aller von Contentpapst in der Datenbank platzierten Tabellen zu finden. Tabellen anderer Programme werden nicht mitgezählt. Diese Größenangaben entsprechen in etwa der Größe der generierten Backup-Datei, sofern Struktur und Daten ausgewählt wurden.

5.19.9. Update aufspielen

Wenn Sie Daten im Format der von Ihnen genutzten Datenbank (standardmäßig MySQL) aufspielen wollen, nutzen Sie das Feld "Update aufspielen" im unteren Bereich des Sicherungs-Modules. Sollten Sie z.B. aus Contentpapst gelöschte Datensätze wiederherstellen wollen, so können frühere Backups in diesem Feld aufgespielt werden. Alternativ kann auch die Backup-Datei hochgeladen werden.



Lassen Sie derartige Updates nur von einem erfahrenen Nutzer durchführen. Unter Umständen riskieren Sie sonst einen Datenverlust, da neue Datensätze mit früheren Versionen (aus dem Backup) überschrieben werden könnten.

5.20. Sprachverwaltung / Mehrsprachigkeit

*Über die Sprachverwaltung in **Contentpapst** können mehrsprachige Seiten aufgebaut und die zugrundeliegenden Sprachdateien bearbeitet werden. Hier erfahren Sie mehr über Leistungsumfang und Möglichkeiten der Mehrsprachigkeit.*

5.20.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=languages	Aufruf über den Administrationsbereich

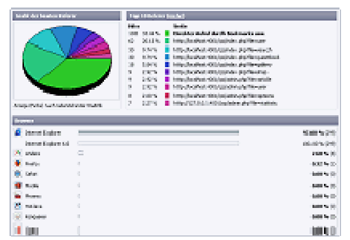
5.21. Statistik

Wenn Sie genauere Informationen über Ihre Besucher erhalten wollen, so sind Sie hier richtig. Sie finden in der Statistik Informationen über die genutzten Browser und Betriebssysteme sowie darüber, aus welchen Ländern Ihre Besucher stammen, welche Seiten wie oft aufgerufen wurden usw.

5.21.1. Allgemeines zu diesem Modul

Das Statistik-Modul zeigt Ihnen im oberen Bereich die Gesamtanzahl der Seitenaufrufe innerhalb der bisherigen Laufzeit der Statistik, sowie die durchschnittlichen Seitenaufrufe pro Tag und Stunde. Daneben befinden sich die Optionen der Statistik. Alle Darstellungen werden in Echtzeit aus der Datenbank generiert.

Die darunter liegende Tabelle stellt die Seitenaufrufe der letzten 30 Tage zusammen mit prozentualen und absoluten Angaben dar. Weiterhin gibt es auf dieser Seite noch die Auflistung der Referer. Wenn die Option "Eigene Seiten als Referer aufzeichnen" aktiviert ist, können Sie hier ablesen, welche Ihrer Seiten am meisten aufgerufen wurden. Die



grafische Darstellung verdeutlicht zusammen mit der Farbkodierung noch einmal den Anteil der einzelnen Seiten. Über den Link "mehr" kann eine ausführliche Auflistung der Referer aufgerufen werden (max. 100 Referer).

Die folgenden Tabellen geben Ihnen Informationen über die genutzten Browser und Betriebssysteme der Besucher. Sollte es sich dabei um Suchmaschinen, Crawler usw. handeln, so werden diese getrennt aufgezeichnet.

Genauere Angaben über die Seitenaufrufe über einen Zeitraum (Monate, Wochentage, Stunden) hinweg erhalten Sie in den entsprechenden Tabellen. Wenn Sie die Herkunftsländer der Besucher aufzeichnen lassen, so steht Ihnen auf einer dritten Seite eine Übersicht über diese Länder zur Verfügung. Weitere Statistiken gibt es für die am häufigsten genutzten Suchbegriffe, die Aufrufe aus Suchmaschinen und Spidern und die Anfragen über die Suchmaschinen sowie die genutzten User-Agents (d.h. die Browserkennzeichnungen usw.).

Wichtige URL's für dieses Modul	
admin.php?file=statistic	Aufruf über den Administrationsbereich

5.21.2. Eigene Seiten als Referer aufzeichnen

Optional ist die Einstellung "Eigene Seiten als Referer aufzeichnen". Diese bewirkt, dass einzelne Seiten Ihrer Webseite als Referer in die Datenbank aufgenommen werden. Dadurch erhalten Sie Informationen über die von Ihren Besuchern am häufigsten aufgerufenen Seiten und Informationen über die Navigationspfade innerhalb der Webseite.

5.21.3. Herkunftsland aufzeichnen

Das Herkunftsland Ihrer Besucher wird anhand der im Browser des Besuchers eingestellten Sprache ermittelt. Ist diese Option aktiviert, so wird für jeden Seitenaufruf das Herkunftsland protokolliert, mit den vorhandenen Einträgen verglichen und in die Datenbank eingetragen. Allerdings verlängert sich durch die Überprüfung der Sprache die Ladezeit der einzelnen Seiten.

5.21.4. Statistik neustarten

Mit einem Klick auf "Statistik jetzt neustarten" lässt sich die Statistik komplett neustarten. Alle bisherigen Einträge werden dadurch zurückgesetzt bzw. aus der Datenbank gelöscht. Wenn Sie vorher ein Backup durchführen wollen, so können Sie dies mit Hilfe des Modules "Sicherung" durchführen. Wählen Sie hier alle Tabellen mit dem Namen "*_stats_*" aus und speichern Sie Struktur und Daten dieser Tabellen auf Ihrer Festplatte.

5.22. Umfragen

Umfragen ermöglichen auf Webseiten eine erweiterte Interaktion zwischen Redaktion und Besuchern. Diese können ihre Meinung kundgeben und so z.B. zu einer Verbesserung der Webseite beitragen. Die Redaktion kann sich genau informieren, wie einzelne Inhalte bei den Besuchen ankommen.

5.22.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=poll	Aufruf über den Administrationsbereich
index.php?file=poll	Aufruf über den öffentlichen Bereich
poll.html	Aufruf über den öffentlichen Bereich (Suchmaschinenfreundliche URL's)

5.22.2. Container: Anzeige der Umfragen auf der Webseite

Dieser Containeraufruf zeigt je nach Einstellung entweder die neueste oder aber eine zufällig ausgewählte Umfrage an. Es sind nur Umfragen mit der Einstellung "Ergebnisse öffentlich zeigen" auf der Webseite sichtbar.

{cp:container module="poll" order="[ID]"}	zeigt eine Umfrage mit einer bestimmten ID an
{cp:container module="poll" order="latest"}	zeigt die neueste Umfrage an
{cp:container module="poll" order="random"}	zeigt eine zufällig ausgewählte Umfrage

Parameter für diesen Container		
order	Auswahl der Umfrage, Werte wie oben sichtbar	optional

5.23. vanillaXML Webshop

Diese Anleitung soll Ihnen die zahlreichen Funktionen von vanillaXML Webshop näher bringen. Dies beinhaltet sowohl den öffentlichen Bereich und den Administrations-Bereich, als auch die direkte Arbeit mit der Entwickler-API von vanillaXML Webshop.

5.23.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=amazon	Aufruf über den Administrationsbereich
index.php?file=amazon	Aufruf über den öffentlichen Bereich

5.23.2. Anpassung der Vorlagen

Fast alle Elemente des vanillaXML Webshop können über Vorlagen an den individuellen Geschmack angepasst werden. Für vanillaXML Webshop kommen dabei zusätzlich zu den normalen Vorlagen des Vorlagen-Sets im Ordner "themes" weitere Vorlagen im Ordner "/modules/amazon/templates/" zum Einsatz:

Von vanillaXML Webshop genutzte Vorlagen	
"browsenode.htm"	Diese Vorlage kommt in der Detailansicht zum Einsatz zur Auflistung der verwandten Kategorien.
"cart.htm"	Der Warenkorb mit der Auflistung aller abgelegten Produkte.
"details.htm"	Detailansicht eines Produktes mit der Anzeige von Tracks, Funktionen etc. je nach dem Katalog des Produktes.
"error_noitems.htm"	Vorlage die angezeigt wird, wenn innerhalb einer Kategorie keine Produkte vorhanden sind.
"error_noresults.htm"	Vorlage die angezeigt wird, wenn bei einer Suchanfrage kein Ergebnis geliefert wird.
"img_full.htm"	Anzeige des vergrößerten Bildes nach Klick auf das Bild in der Detailansicht eines Produktes.
"list.heavy.htm"	Vorlage für die Auflistung von Produkten mit erweiterten Angaben.
"list.lite.htm"	Vorlage für die Auflistung von Produkten mit einfachen Angaben (Standardvorlage für die Auflistung).
"node_show.htm"	Anzeige einer einzelnen Kategorie mit der Anzahl der enthaltenen Produkte etc.
"reviews.htm"	Auflistung der einzelnen Kundenmeinungen zu einem Produkt.
"search_full.htm"	Vorlage für die Darstellung des Suchergebnisses (Übersicht über die einzelnen Kataloge).
"search_show.htm"	Vorlage für die Darstellung des Suchergebnisses (Detailansicht einer einzelnen Suche).
"similiar.htm"	Auflistung der verwandten Produkte in der Detailansicht eines Produktes.
"sitemap_standard.htm"	Standardvorlage für die Erstellung einer Sitemap über das Sitemap-Tool.

5.23.3. Eigene Seiten mit Produktansichten etc. erstellen

Über die im System integrierten Container, die Arbeit mit Vorlagen etc. ist es nach kurzer Eingewöhnung möglich, eigene Seiten mit beliebigen Zusammenstellungen an Produkten, Suchabfragen, Kategorieinformationen etc. zu erstellen.

Die Beispiele für die Verwendung sind vielfältig, u.a. die Erstellung von themenspezifischen Sonderseiten, z.B. zu "Reisen nach Italien", "Kochen & Backen" oder auch passend zur "Fussball-Europameisterschaft". Bei geschicktem Einsatz der Container können Sie so auch ganz neue Zielgruppen ansprechen, die normalerweise nicht in Ihrem Shop einkaufen würden.

Mehr Informationen über die Verwendung von eigenen Seiten erhalten Sie auch unter "Vorlagen: Statische Seiten in Contentpapst einbinden". Zudem sollten Sie sich die einzelnen für den vanillaXML Webshop zur Verfügung stehenden Container genauer anschauen.

Sobald Sie dies vorgenommen haben, kann die Realisierung einer ersten Sonderseite beginnen. Entscheiden Sie sich, ob diese innerhalb des allgemeinen Layouts des Shops oder als eigenständige HTML-Seite dargestellt werden soll. Dementsprechend muss dann auch der Aufbau der Seite ausfallen (d.h. bei einer kompletten Seite inkl. HTML-Kopf etc.). In diese Seite integrieren Sie dann die {cp:tags} zur Anzeige von einzelnen Produkten, von ganzen Suchabfragen etc. sowie die gewünschten Grafiken und Texte. Einige Beispiele für Tags:

```
{cp:container module="amazon" type="searchbrowsenode" query="10925241" limit="5" cache="86400"
sort="+daterank" template="modules/amazon/templates/list.lite.htm"}
// zeigt die neuesten 5 Produkte im Katalog "Garten & Freizeit" (Browse-iD 10925241) über die Standard-Vorlage an
```

```
{cp:container module="amazon" type="products" asin="B0001VQI2M" catalog="garden" template="special.htm"}
// zeigt das Produkt mit der ASIN "B0001VQI2M" aus dem Katalog "Garten & Freizeit" über die Vorlage "special.htm" an
```

Auch Fremdinhalte wie Newsticker, RSS/RDF-Feeds etc. können in diesen Seiten eingebunden werden. Die detaillierten Informationen finden Sie in den Beschreibungen der einzelnen {cp:tags}.

5.23.4. Allgemeines zum Einsatz der Container

Die Container können Sie innerhalb der vorhandenen Vorlagen einsetzen oder Sie platzieren diese in einer beliebigen HTML-Datei, die Sie unterhalb des Hauptordners ablegen und dann über "index.php?static=[PFAD ZUR DATEI.HTM]&clean=yes" aufrufen. Über diesen Weg ist es möglich, Produkte von Amazon z.B. innerhalb anderer Texte, einem Forum etc. anzuzeigen. An ein festes Layout sind Sie durch die freie Gestaltung der HTML-Vorlagen nicht gebunden.

5.23.5. Container: Angepasste Navigationen nutzen

Neben dem Container für die Einblendung der Standardnavigation gibt es auch die Möglichkeit, eine den eigenen Vorstellungen angepasste Navigation über den Container "customnavigation" zu erstellen:

```
{cp:container module="amazon" type="customnavigation" browsenode="..." template="..." [template_subnodes="..."]
[order="..."] [dir="..."] [block="..."]}
```

Oben sehen Sie den allgemeinen Aufbau des Containers mit den verfügbaren Parametern. Als "browsenode" wird dabei stets die Browse-ID aus dem Kategorien-Editor angegeben, welche als Startpunkt verwendet werden soll. Diese muss

jedoch zuvor über <?php base64_encode("__BROWSE_ID__"); ?> verschlüsselt werden.

Dazu kommt die Angabe der zu nutzenden Vorlage mit "template" sowie mit "template_subnodes" eine Vorlage, die nur Verwendung findet, wenn Unterkategorien vorhanden sind. Innerhalb der Vorlagen kann mit den Tags "{cp:amazon:navigation:title}", "{cp:amazon:navigation:number}" und "{cp:amazon:navigation:browsenode}" gearbeitet werden, um die Inhalte einzusetzen.

Über den Parameter "order" wird die Ausgabe geordnet (mit den Werten "number", "browsenode" oder "title"), mit "dir" wird die Richtung der Ordnung vorgegeben ("asc" für aufsteigend oder "desc" für absteigend) und mit dem Parameter "block" kann eine kommaseparierte Liste an Browse-ID's übergeben werden, welche in der Navigation nicht auftauchen sollen.

5.23.6. Container: Beliebteste Kategorien

Um Ihren Kunden weitere Kaufempfehlungen zu geben, können Sie die aus den Statistiken im Administrationsbereich bekannte Auflistung der beliebtesten Kategorien auch im öffentlichen Bereich über einen Container anzeigen lassen:

```
{cp:container module="amazon" type="favouritecategories" [limit="5"]}
```

Über den Parameter "limit" ist eine Eingrenzung der maximalen Kategorieanzahl möglich. Ist "limit" nicht gesetzt, wird automatisch nur eine Auflistung der ersten 10 Kategorien ausgeliefert. Um durch diesen Container nicht die Ladezeit der Webseite zu erhöhen, sollten Sie diesen über den Parameter "cache" zusätzlich für eine längere Zeit zwischenspeichern (z.B. "86400" Sekunden für einen Tag).

5.23.7. Container: Beliebteste Produkte

Um Ihren Kunden weitere Kaufempfehlungen zu geben, können Sie die aus den Statistiken im Administrationsbereich bekannte Auflistung der beliebtesten Produkte auch im öffentlichen Bereich über einen Container anzeigen lassen:

```
{cp:container module="amazon" type="favouriteproducts" [limit="5"]}
```

Über den Parameter "limit" ist eine Eingrenzung der maximalen Produktanzahl möglich. Ist "limit" nicht gesetzt, wird automatisch nur eine Auflistung der ersten 10 Produkte ausgeliefert. Um durch diesen Container nicht die Ladezeit der Webseite zu erhöhen, sollten Sie diesen über den Parameter "cache" zusätzlich für eine längere Zeit zwischenspeichern (z.B. "86400" Sekunden für einen Tag).

5.23.8. Container: Besondere Eigenschaften (bei Elektronik, DVD und Zeitschriften)

Vergleichbar zu den Plattformen bei Software etc. gibt es für die Kataloge Elektronik, DVD und Zeitschriften zusätzliche Informationen in Form der detaillierten Eigenschaften (beispielsweise die maximale Auflösung bei einem Computer-Monitor). Für vanillaXML Webshop steht zur Einblendung dieser Informationen ein weiterer Container zur Verfügung:

```
{cp:container module="amazon" type="features" file="abc.xml"}
// Besondere Eigenschaften, generiert aus der Datei abc.xml
```

Damit die richtige Datei als Datenquelle genutzt wird, muss diese über den Parameter "file" angegeben werden. Allgemein sollte dieser Container nicht manuell gesetzt werden, sondern wird direkt in der Detailansicht eines Produktes eingesetzt. Auch diese Informationen können wieder per CSS angepasst werden.

5.23.9. Container: Letzte Suchanfragen

Um Ihre Besucher bei der Suche ein wenig zu inspirieren, bietet es sich an, die letzten Suchanfragen an geeigneter Stelle einzublenden. Durch einen Klick auf die einzelnen Begriffe ist dann schnell ein Wechsel zur jeweiligen Ergebnisseite der Suche möglich. Häufig nachgefragte Produkte können so im Absatz noch weiter gestärkt werden.

Über die Parameter "limit" und "catalog" kann zusätzlich die Ausgabe beeinflusst werden. "limit" gibt hierbei die maximale Anzahl der angezeigten Suchanfragen an, "catalog" begrenzt hingegen die Ausgabe auf einen bestimmten Produktkatalog wie z.B. Musik (Katalog "music").

```
{cp:container module="amazon" type="last_searches" limit="10" cache="180"}
// Beispiel mit 10 Suchanfragen und aktiviertem Cache
```

5.23.10. Container: Musiktitel (nur bei Musik)

Nur beim Katalog Musik stehen bei vielen Produkten genaue Informationen über die einzelnen Musiktitel zur Verfügung. Diese Informationen sind für Ihre Besucher natürlich besonders wichtig. Für vanillaXML Webshop steht zur Einblendung dieser Informationen ein weiterer Container zur Verfügung:

```
{cp:container module="amazon" type="tracks" file="abc.xml"}
// Besondere Eigenschaften, generiert aus der Datei abc.xml
```

Damit die richtige Datei als Datenquelle genutzt wird, muss diese über den Parameter "file" angegeben werden. Allgemein sollte dieser Container nicht manuell gesetzt werden, sondern wird direkt in der Detailansicht eines Produktes eingesetzt. Auch diese Informationen können wieder per CSS angepasst werden.

5.23.11. Container: Navigation einblenden (Produktkategorien)

Die Navigation aller Produktkategorien wird von vanillaXML Webshop automatisch in Abhängigkeit der gerade geöffneten Seite und der ausgewählten Sprache aufgebaut. Die Integration der Navigation in die Webseite geschieht über einen einfachen {cp:tag} in einer beliebigen Vorlage:

```
{cp:container module="amazon" type="navigation"}
```

5.23.12. Container: Plattformen (bei Software und Spielen)

Für die Kataloge Software und Computer- & Videospiele werden zusätzliche Informationen über die unterstützten Plattformen, d.h. die Betriebssysteme geliefert. Für vanillaXML Webshop steht zur Einblendung dieser Informationen ein weiterer Container zur Verfügung:

```
{cp:container module="amazon" type="plattforms" file="abc.xml"}
// Produkt-Plattformen, generiert aus der Datei abc.xml
```

Damit die richtige Datei als Datenquelle genutzt wird, muss diese über den Parameter "file" angegeben werden. Allgemein sollte dieser Container nicht manuell gesetzt werden, sondern wird direkt in der Detailansicht eines Produktes eingesetzt.

Auch diese Informationen können wieder per CSS angepasst werden. Soweit vorhanden wird für die Plattformen (u.a. Windows XP, Windows 2000, Gameboy, Playstation 2, X-Box etc.) das passende Logo daneben eingeblendet.

5.23.13. Container: Produktinformationen abrufen

Wenn Sie einzelne Produkte innerhalb anderer Seiten dynamisch einbinden möchten, können Sie dies direkt über einen {cp:tag} als Container realisieren. Der Container ermöglicht es Ihnen zusätzlich die Produkte beliebig oft in unterschiedlichen Layouts darzustellen.

```
{cp:container module="amazon" type="products" asin="B0000AJ595" catalog="music" template="abc.htm"}
// Musik-CD Dido "Life For Rent" dargestellt über die Vorlage "abc.htm".
```

Wenn im obigen {cp:tag} der Parameter "template" nicht angegeben wird, verwendet vanillaXML Webshop automatisch die Vorlage "/modules/amazon/templates/details.htm". Für eine manuelle Anpassung der Abfrage über die Programmiersprache PHP empfiehlt sich ein Blick in die Datei "/modules/amazon/container/products.php".

5.23.14. Container: Suchabfragen durchführen

Besonders interessant - gerade beim Einsatz von vanillaXML Webshop als Contentpapst-Modul - sind die Container für die Durchführung von Suchabfragen. In einem Text zum Thema Tauchsport bspw. könnte direkt ein Container platziert werden, der thematisch passende Bücher bei Amazon heraussucht und diese dort einblendet. Der Absatz lässt sich durch derartige Verknüpfungen stark steigern.

Im Folgenden werden eine Reihe von Containern vorgestellt, jeweils mit einem Beispiel. Die Parameter "query" (bspw. das Suchwort), "catalog" (der Produktkatalog), "complexity" (Komplexität der Abfrage, "lite" oder "heavy"), "sort" (Sortierung) und "page" (aktuelle Seite) sind jeweils allgemeingültig. Über den Parameter "limit" muss zusätzlich die gewünschte Anzahl der Suchergebnisse angegeben werden. Maximal sind dies 10 Suchergebnisse pro Seite.

Je nach Einstellung des Parameters "complexity" wird die Vorlage "/modules/amazon/templates/list.heavy.htm" oder die Vorlage "/modules/amazon/templates/list.lite.htm" genutzt. Alternativ kann über den Parameter "template" eine eigene Vorlage eingesetzt werden.

Suche nach Schlüsselwörtern:

```
{cp:container module="amazon" type="searchkeyword" query="Tauchen"}
// einfache Suche nach Büchern mit dem Begriff "Tauchen"

{cp:container module="amazon" type="searchkeyword" query="Fussball" catalog="vhs" sort="+title" page="2"}
// Suche nach Fussball-Videos, Seite 2 geordnet nach dem Titel
```

Suche nach Autoren:

```
{cp:container module="amazon" type="searchauthor" query="Dan Brown"}
// einfache Suche nach Büchern mit dem Autor "Dan Brown"
```

Suche nach ähnlichen Produkten:

```
{cp:container module="amazon" type="searchsimilarity" query="B00005Q66T"}
// Suche nach ähnlichen Produkten zum Produkt mit der ASIN "B00005Q66T"
```

Suche nach Regisseuren / Produktionsleitung:

```
{cp:container module="amazon" type="searchdirector" query="Andrew Stanton"}
// einfache Suche nach Filmen mit dem Regisseur "Andrew Stanton"
```

Suche nach Künstlern:

```
{cp:container module="amazon" type="searchartist" query="Madonna" catalog="music"}
// einfache Suche nach Musik-CD's von "Madonna"
```

Suche nach Herstellern:

```
{cp:container module="amazon" type="searchmanufacturer" query="Sony" catalog="electronics"}
// Suche "Sony" im Katalog "Elektronik"
// unterstützt werden hier "electronics", "kitchen", "videogames", "software", "photo" und "pc"
```

Suche nach Schauspielern:

```
{cp:container module="amazon" type="searchactor" query="Renée Zellweger" catalog="dvd"}
// einfache Suche nach DVD's mit der Schauspielerin "Renée Zellweger"
```

Suche nach Produktkategorien:

```
{cp:container module="amazon" type="searchbrowsenode" query="288100"}
// Produkte in der Kategorie "Fachbücher" werden hier gesucht
```

Suche nach Produkten per UPC (Unique Product Code):

```
{cp:container module="amazon" type="searchupc" query="B0000CFW87"}
// Suche nach einem UPC (momentan nicht in Deutschland möglich)
```

Aufgrund der Ladezeit, die teilweise durch die Amazon-Server entstehen kann, empfiehlt es sich über den Parameter "cache" die einzelnen Suchabfragen in den Zwischenspeicher legen zu lassen (bspw. "86400" für 24 h).

5.23.15. Container: Warenkorb mit Produktinformationen

Über die Einblendung des Warenkorbs als Container haben Ihre Besucher stets im Blick, welche Produkte bereits dort abgelegt wurden und für die Bestellung bereitgestellt sind. Der Container kann wie üblich innerhalb einer beliebigen Vorlage platziert werden. Der Aufbau wird direkt von vanillaXML Webshop realisiert, kann aber mittels CSS angepasst werden.

```
{cp:container module="amazon" type="cart"}
```

Im Container werden zu jedem Produkt Menge und Produktname eingeblendet, sowie der Gesamtpreis (ohne event. Versandgebühren). Sollten keine Produkte im Warenkorb liegen wird eine Alternativmeldung ausgegeben.

5.23.16. Suchmaschinenfreundliche URL's aktivieren

Der vanillaXML Webshop ist bereits standardmäßig für optimale Suchmaschinenfreundlichkeit vorbereitet. Dazu gehört u.a. auch die Simulation von statischen Seiten mit der Endung ".html" oder von Ordnern. Dies wird über eine Erweiterung des Apache-Webservers mit dem Namen "mod_rewrite" möglich. Ausführliche Informationen finden Sie u.a. in "Systemkomponenten / Module > Einstellungen > Suchmaschinenfreundliche URL's" und "FAQ: Kann ich meine Seite gezielt für Suchmaschinen optimieren?".

Zusätzlich zu den dort genannten Schritten zur Aktivierung der suchmaschinenfreundlichen URL's benötigt der vanillaXML Webshop einige weitere Regeln, die nach der Aktivierung in der Datei ".htaccess" zu platzieren sind (unterhalb der standardmäßigen RewriteRules):

```
RewriteRule ^(de)/([a-z_]+)/([0-9]+)/([^/]+)/ http://[DOMAIN]/index.php?
file=amazon&mode=catalog&catalog=$2&page=$3&sort=$4&language=$1&mod_rewrite=1 [L]

RewriteRule ^(de)/([a-z_]+)/([0-9]+)/ http://[DOMAIN]/index.php?
file=amazon&mode=catalog&catalog=$2&page=$3&language=$1&mod_rewrite=1 [L]

RewriteRule ^(de)/search/([a-z_]+)/([^/]+)/([0-9]+)/([^,]+),([a-z]+) http://[DOMAIN]/index.php?
file=amazon&mode=search&q=$3&form[catalog]=$2&form[sort]=$5&form[complexity]
=$6&page=$4&language=$1&mod_rewrite=1 [L]

RewriteRule ^(de)/search/([a-z_]+)/([^/]+)/([0-9]+)/ http://[DOMAIN]/index.php?
file=amazon&mode=search&q=$3&form[catalog]=$2&page=$4&language=$1&mod_rewrite=1 [L]

RewriteRule ^(de)/(search|cart)/ http://[DOMAIN]/index.php?file=amazon&mode=$2&language=$1&mod_rewrite=1
[L]

RewriteRule ^(de)/(author|artist)/([^/]+)/ http://[DOMAIN]/index.php?file=amazon&mode=search&form[type]
=$2&q=$3&language=$1&mod_rewrite=1 [L]

RewriteRule ^(de)/([a-z_]+)/details/([A-Za-z0-9]+)/ http://[DOMAIN]/index.php?
file=amazon&mode=details&number=$3&catalog=$2&language=$1&mod_rewrite=1 [L]

RewriteRule ^(de)/([a-z_]+)/([A-Za-z0-9=]+)/([^/]+)/([0-9]+)/([^/]+)/ http://[DOMAIN]/index.php?
file=amazon&mode=node&number=$3&page=$5&sort=$6&catalog=$2&language=$1&mod_rewrite=1 [L]

RewriteRule ^(de)/([a-z_]+)/([A-Za-z0-9=]+)/([^/]+)/([0-9]+)(/?) http://[DOMAIN]/index.php?
file=amazon&mode=node&number=$3&page=$5&catalog=$2&language=$1&mod_rewrite=1 [L]

RewriteRule ^(de)/([a-z_]+)/([A-Za-z0-9=]+)/ http://[DOMAIN]/index.php?
file=amazon&mode=node&number=$3&catalog=$2&language=$1&mod_rewrite=1 [L]
```

```
RewriteRule ^(de)/([a-z_]+)/ http://[DOMAIN]/index.php?
file=amazon&mode=catalog&catalog=$2&language=$1&mod_rewrite=1 [L]
```

"[DOMAIN]" ersetzen Sie bitte durch Ihren vollen Domainnamen sowie gegenbenenfalls den Pfad zum Verzeichnis, in dem sich vanillaXML Webshop befindet.

Die URL's werden dann vom System meist automatisch im neuen Format generiert. Um auch die Vorlagen in den Ordnern "/modules/amazon/templates/" und "/themes/frontend/" zu ändern, tauschen Sie die normalen Vorlagen mit den "mod_rewrite"-Vorlagen aus (z.B. wird "cart.htm" ersetzt durch "modrewrite_cart.htm", durch Umbenennen). Die in den Vorlagen bisher genutzten dynamischen URL's werden dann automatisch durch die statischen URL's ersetzt.

Zusätzlich müssen Sie - wenn Sie den vanillaXML Webshop mit "mod_rewrite" in einem Unterverzeichnis betreiben - in den Vorlagen bei der Links jeweils noch Ihr Unterverzeichnis einfügen (nur bei den internen Links, d.h. nicht bei Links die direkt zu Amazon etc. gehen) bzw. Sie geben das Unterverzeichnis unter "Einstellungen > Allgemeine Einstellungen" an.

5.23.17. Übersicht über die Methoden der Entwickler-API

Die Entwickler-API ist die direkte Schnittstelle von vanillaXML Webshop zu Amazon. Auf der Basis einiger grundlegenden Methoden werden Daten von Amazon geholt, verarbeitet und ausgeben. Mit ein wenig Eingewöhnung werden Sie schnell eigene Klassen ableiten können, die die von Ihnen benötigten Funktionen optimal kombinieren.

Aktuell genutzte Methoden der Entwickler-API	
cache_images()	speichert ein Bild lokal ab
check_availability()	überprüft, ob das Produkt auch verfügbar ist bzw. nachbestellt werden kann
check_type()	überprüft die Gültigkeit des Typs ("lite"/"heavy")
create_product_list()	generiert mittels einer Vorlage eine Produktauflistung (z.B. für Übersichtsseiten)
create_review_list()	erstellt eine Liste der Rezensionen eines Produkts mittels einer Vorlage
create_similiar_list()	erstellt eine Liste der verwandten Produkte zu einem oder mehreren Produkten
db_query()	hält die Verbindung zur Datenbank (für die Statistik)
form_add_items()	Formular zum Hinzufügen von Produkten
form_add_marketplace ()	Formular zum Hinzufügen von Marktplatz-Produkten
form_add_wishlist()	Formular zum Hinzufügen von Produkten zur Wunschliste
get_affiliate_id()	Affiliate-ID erhalten
get_affiliate_link()	Liefert anhand von $type einen Affiliate-Link
get_asin_link()	Generiert Link zu Produktseite mit Affiliate-ID anhand einer ASIN (mit $direct direkt zur Produktseite)
get_browse_node_link()	generiert Link zu einer bestimmten Kategorie mit Affiliate ID
get_content()	holt den XML-Inhalt aus einer Suchabfrage ab und speichert diesen
get_info()	gezielt eine einzelne Information zu einem Produkt abfragen
get_keyword_link()	Generiert Link zu Produktsuche mit verschiedenen Parametern
get_link()	Generiert einen beliebigen Link zu Amazon (immer ab "tg/browse/..." angeben)
get_locality()	Landeskennung erhalten
get_search_url()	setzt die URL für die Abfragen zusammen
get_subscription_id()	Subscription-ID erhalten
get_rewrite_name()	stellt für get_url den Namen im mod_rewrite-Format zusammen
get_url()	erzeugt die URL zu einem Artikel / einer Kategorie
local_data()	liefert die landesspezifischen Einstellungen zu Amazon
num_results()	gibt die Anzahl der Ergebnisse einer Suchabfrage zurück
reset()	Setzt die temporären Daten zurück (intern genutzt)
save()	Speichert je nach Parameter XML oder den Inhalt als Datei ab (intern genutzt)
search_actor()	Suche nach Schauspielern (liefert passende DVDs)
search_artist()	Suche nach Künstlern (liefert passende CDs)
search_asin()	Details über ein bestimmtes Produkt heraussuchen
search_author()	Suche nach Autoren (liefert passende Bücher)
search_blended()	Suche über alle Kategorien hinweg
search_browse_node()	Suche nach Produktkategorien
search_director()	Suche nach Produktionsleitung (liefert passende DVDs)
search_exchange()	Finds a given exchange
search_isbn()	Greift auf search_asin zurück
search_keyword()	Führt eine Stichwort-Suche durch
search_listmania()	Sucht Listen nach Listmania-IDs
search_manufacturer()	Suche nach Hersteller
search_power()	Erweiterte Suche ("PowerSearch")
search_seller()	Findet Produkte zu einer Seller-ID
search_seller_profile()	Liefert das Profil zu einer Seller-ID
search_similarity()	Sucht ähnliche Produkte zu einer ASIN
search_upc()	Suche nach UPC (nur in Musik verfügbar)
search_wishlist()	Sucht Wunschlisten nach ID
set_affiliate_id()	Affiliate-ID festlegen
set_cache_time()	legt Cache-Zeit in Sekunden fest (anhand der Amazon-Regeln)

set_locality()	Landeskennung festlegen
set_price()	Preisgrenzen festlegen (noch nicht überall bei Amazon möglich)
set_options()	Einzelne Einstellungen lassen sich zentral über diese Methode anpassen.
set_show_all()	Zeigt alle Produkte an, egal ob diese verfügbar sind oder nicht.
set_subscription_id()	Subscription-ID festlegen

5.23.18. Fehlercodes / -meldungen

vanillaXML Webshop liefert bei verschiedenen Methoden der API Fehlercodes / -meldungen zurück. Die wichtigsten dieser Meldungen sind hier aufgelistet, zusammen mit den jeweiligen Methodennamen:

0	Keine Affiliate-ID angegeben. (get_affiliate_link/get_affiliate_id/set_affiliate_id)
1	Keine Länderkennung angegeben. (get_locality/set_locality)
2	Notwendige Angaben fehlen. (get_subscription_id/set_subscription_id/set_cache_time/get_browse_node_link/get_keyword_link)
3	Keine Amazon-Webservices in diesem Land verfügbar.
4	Dieser Modus existiert nicht.
5	Status kann nur "open" oder "closed" sein.
6	Daten konnten erfolgreich geholt werden.

5.24. Webservices

Über die Webservices können verschiedene Systeme effizient miteinander verknüpft werden, indem die Datei über standardisierte und meist auf XML basierende Dateiformate ausgetauscht werden. ***Contentpapst*** *bietet diese Möglichkeiten bereits in vielen Modulen.*

5.24.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=webservices	Aufruf über den Administrationsbereich
index.php?file=webservices&...	Download der ".klip"-Datei (Steuerungsdatei für KlipFolio)
index.php?file=webservices&...	Download der letzten News (Inhaltsdatei für KlipFolio)
index.php?file=webservices&...	Ausgabe der News im XML-Format (RDF 0.9)

5.24.2. Container: Feeds von externen Webseiten in die eigene Webseite einbinden

Viele Webseiten und Blogs, welche auf Content-Management-Systemen basieren, bieten mittlerweile Inhalte für andere Webseiten an. Auch bei Contentpapst ist dies als Funktion integriert. Mit {cp:container module="webservices"} können diese externen Inhalte in den Formaten XML, RSS und RDF importiert und angezeigt werden. Die Gestaltung ist mittels Vorlagen realisierbar und alle Inhalte können zwischengespeichert werden, so dass nicht bei jedem Aufruf der Seite ein erneutes Herunterladen der externen Datei notwendig wird.

Der Containeraufruf {cp:container module="webservices"} besteht im Wesentlichen aus einer Reihe von Angaben, welche Typ und Ort der anzuzeigenden Datei beschreiben:

```
{cp:container module="webservices" type="rdf" file="http://www.heise.de/newsticker/heise.rdf "
template="templates/heise.htm" limit="10"}
```

Der oben sichtbare Containeraufruf gibt die ersten zehn Artikel des RDF-Feeds von http://www.heise.de/ aus. Die Gestaltung der Artikel wird über die neue Vorlage "templates/heise.htm" realisiert, u.a. mit den Tags {cp:webservices:link} und {cp:webservices:title}.

Parameter für diesen Container		
file	Externe Datei mit den Inhalten	erforderlich
limit	Anzahl der insgesamt angezeigten Einträge	optional
template	Vorlage für die Gestaltung	optional
type	Typ der Datei ("RSS","RDF","XML")	erforderlich

5.24.3. Newsfeed für die Software "KlipFolio"

Für das praktische Windows-Programm KlipFolio von Serence, Inc. beinhaltet Contentpapst ein eigenes Newsfeed. Einmal eingerichtet, können Nutzer von KlipFolio damit die News/Artikel Ihrer Webseite mit KlipFolio abrufen. Dies hat den Vorteil, dass man immer über Updates auf Ihrer Webseite informiert wird. Sorgen Sie mit dem Newsfeed für KlipFolio für einen dauerhaften Besucheranstrom. Den Newsfeed richten Sie folgendermaßen ein:



- laden Sie die neueste Version von der **Webseite des Herstellers** herunter
- starten Sie die **Installationsroutine** und speichern Sie anschließend die Datei newsfeed.klip auf Ihrer lokalen Festplatte
- erstellen Sie eine Grafik (GIF/JPEG) im Format 234x60 Pixel und speichern Sie diese mit dem Dateinamen "klipbanner" im Ordner "images/".
- weiterhin erstellen Sie ein Icon (GIF/JPEG) im Format 16x16 Pixel und speichern Sie dieses als "klipicon" im Ordner

"images/".

- rufen Sie die **KlipFolio-Webseite** auf und melden Sie sich als Publisher an
- hier können Sie nach dem Einloggen die auf Ihrer lokalen Festplatte gespeicherte Datei newsfeed.klip sowie die Icon-Grafik hochladen
- nach der Überprüfung mittels des KlipFolio-Programmes können Sie Ihren Klip freischalten

5.25. Werbung

Schalten Sie Werbeeinblendungen auf Ihrer Webseite und verdienen Sie. Sie können verschiedene Werbeformen parallel laufen lassen, indem Sie die entsprechenden {cp:tags} in die gewünschten Vorlagen oder Artikel einbauen. Neben den Standardformaten für Bannerwerbung sind auch Werbeschaltungen als Flashfilm oder per Text/HTML möglich.

5.25.1. Allgemeines zu diesem Modul

Wichtige URL's für dieses Modul	
admin.php?file=ads	Aufruf über den Administrationsbereich
index.php?file=ads	Aufruf über den öffentlichen Bereich (Kundenzugang)
ads.html	Aufruf über den öffentlichen Bereich (Kundenzugang, Suchmaschinenfreundliche URL's)

5.25.2. Werbekunden einrichten / verwalten

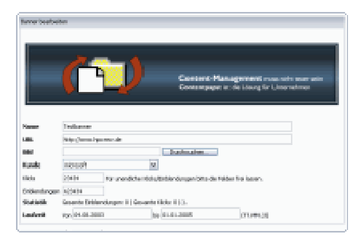
Jedes Werbemittel wird einem Kunden zugewiesen, welcher zuvor getrennt eingerichtet werden muss. Für die Einrichtung eines neuen Kunden ist u.a. die Angabe von Name, Benutzername und Passwort notwendig. Optional sind der Name einer Kontaktperson, sowie eine Email-Adresse. Benutzername und Passwort dienen dabei gleichzeitig dem Kunden für den Abruf der Werbemittelstatistik.

Auch nachdem dem Kunden Werbemittel zugeordnet wurden, lassen sich über den Administrationsbereich jederzeit die Daten des Kunden bearbeiten. Sollten dem Kunden keine Werbemittel mehr zugeordnet sein, so kann dieser wieder gelöscht werden.

5.25.3. Kundenzugang

Ihre Kunden erhalten einen separaten Login-Bereich, unter welchem Sie ihre Benutzerdaten ändern und die aktuelle Statistik ihrer Werbeeinblendungen abrufen können. Angezeigt werden nur die Werbemittel des Kunden selbst, so dass dieser keinen Einblick auf fremde Werbemittel erhält. Der Kunde kann sich auf der Seite "index.php?file=ads" einloggen. Hierfür benötigt er lediglich Benutzername und Passwort.

5.25.4. Werbemittel einstellen



Als Werbemittel sind neben verschiedenen Bannerformaten (468x60, 300x300, Skyscraper usw.) auch Text/HTML und Flashfilme zulässig. Diese werden zusammen mit einer URL und einem Namen zentral im Modul eingestellt.

Jedes Werbemittel wird dabei einem Kunden zugewiesen, damit dieser den Erfolg seiner Werbemittel kontrollieren kann. Die Anzeige der Werbemittel auf der Webseite kann in der Laufzeit (Format TT.MM.JJ) sowie in der Anzahl von Klicks und Einblendungen begrenzt werden.

Wenn die Anzeige durch Einblendungen und Klicks begrenzt wird, so wird bei Erreichen der maximalen Einblendungen das Werbemittel nicht mehr angezeigt, auch wenn noch Klicks offen sind. Die Anzeige auf der Webseite wird durch den {cp:tag} "{cp:container module="ads"}" in Kombination mit der "type"-Angabe gesteuert.

5.25.5. Container: Anzeige der Werbemittel auf der Webseite

Positionieren Sie die Werbemittel frei auf der Webseite, indem Sie diesen Containeraufruf innerhalb der Vorlagen einbauen. Mittels des Parameters "type" kann eine spezielle Werbeform ausgewählt werden, Wird der Parameter weggelassen, können verschiedene Werbeformen an dieser Position eingeblendet werden. Folgende Tags sind möglich:

{cp:container module="ads" type="0"}	Freie Größe	nicht festgelegt
{cp:container module="ads" type="2"}	Full Banner	(468x60 px)
{cp:container module="ads" type="3"}	Half Banner	(234x60 px)
{cp:container module="ads" type="4"}	Vertical Banner	(120x240 px)
{cp:container module="ads" type="6"}	Button	(120x60 px)
{cp:container module="ads" type="7"}	Square Button	(125x125 px)
{cp:container module="ads" type="8"}	Micro Button	(88x31 px)
{cp:container module="ads" type="10"}	Skyscraper	(120x600 px)
{cp:container module="ads" type="11"}	Wide Skyscraper	(160x600 px)
{cp:container module="ads" type="12"}	Medium Rectangle	(300x250 px)
{cp:container module="ads" type="16"}	Large Rectangle	(336x280 px)
{cp:container module="ads" type="17"}	Leaderboard	(728x90 px)
{cp:container module="ads" type="18"}	Small Square	(200x200 px)
{cp:container module="ads" type="19"}	Square	(250x250 px)
{cp:container module="ads" type="20"}	Half Page Ad	(300x600 px)
{cp:container module="ads" type="21"}	Rectangle	(180x150 px)

{cp:container module="ads" type="13"}	Flash (SWF)	beliebige Größe
{cp:container module="ads" type="14"}	HTML-Code	beliebige Größe
{cp:container module="ads" type="15"}	Quicktime (Apple)	beliebige Größe

Parameter für diesen Container		
adspace	Gewünschter Werbebereich, per Kommata getrennt sind auch mehrere Werbebereiche möglich	optional
limit	Anzahl der maximal sichtbaren Werbemittel (standardmäßig "1")	optional
type	Werbeform, Werte wie oben sichtbar	optional
unlimited	Anzeige unabhängig von der Laufzeit, Wert "1" oder "yes"	optional

6. Erweiterung des Systems per Entwickler-API

*Mit der Entwickler-API gibt es einen Überblick über die öffentlichen Funktionen, welche bei der Entwicklung eigener Module für **Contentpapst** genutzt werden können. Dies sind Funktionen zur Interaktion mit anderen Modulen sowie auch mit externen Resourcen. Beachten Sie insbesondere die "Entwickler-Informationen" für die Entwicklung komplett neuer Module.*

6.1. Übersicht

6.1.0. Allgemeine Informationen

Zusätzlich zu den standardmäßigen Funktionen kann das Content-Management-System **Contentpapst** auch um weitere Funktionen auf der Basis der Scriptsprache PHP erweitert werden. Damit die bereits in **Contentpapst** enthaltenen Funktionen nicht für die eigenen Erweiterungen erneut programmiert werden müssen, können die internen Funktionen auch über eine API direkt angesprochen werden.

Folgende PHP-Dateien können in eigene Erweiterungen eingebunden werden:

- **/functions/cp.css.php**
  Funktionen für die Arbeit mit CSS-Dateien.
- **/functions/cp.date.php**
  Funktionen für die Arbeit mit Datumsangaben (z.B. Validierung einer Datumsangabe).
- **/functions/cp.files.php**
  Funktionen für die Arbeit mit Dateien und Ordnern.
- **/functions/cp.ftp.php**
  Funktionen für den Austausch von Dateien per FTP.
- **/functions/cp.graph.php**
  Funktionen für die Erstellung von Grafiken (Spezialfunktionen für Statistiken).
- **/functions/cp.image.php**
  Funktionen für die Erstellung von Grafiken (Grundfunktionen für normale Grafiken).
- **/functions/cp.iso.php**
  Verschiedene Arrays mit Informationen zu ISO-Codes (u.a. internationale Länderkürzel).
- **/functions/cp.mail.php**
  Funktionen für die Erstellung von (HTML-)Emails.
- **/functions/cp.mime.php**
  Umfangreiches Array mit Informationen zu MIME-Typen.
- **/functions/cp.misc.php**
  Vermischte Funktionen, u.a. für die Arbeit mit vers. Zeichensätzen.
- **/functions/cp.navigation.php**
  Funktionen für die Erstellung von Navigationen.
- **/functions/cp.pdf.php**
  Funktionen zur Erstellung von PDF-Dateien.
- **/functions/cp.rtf.php**
  Funktionen zur Erstellung von RTF-Dateien.
- **/functions/cp.search.php**
  Funktionen für die Volltextsuche des Systems.
- **/functions/cp.session.php**
  Funktionen für die Speicherung von temporären Benutzerinformationen in Sessions.
- **/functions/cp.statistic.php**
  Funktionen für die Statistik des Systems.
- **/functions/cp.string.php**
  Funktionen für die Arbeit mit Texten ("Strings").
- **/functions/cp.template.php**
  Funktionen für die Verarbeitung von Vorlagen (inkl. der {cp:tags}).
- **/functions/cp.url.php**
  Funktionen für die Arbeit mit URL's (z.B. Verlegung in einzelne Bestandteile).
- **/functions/cp.user.php**
  Funktionen rund um die Benutzerverwaltung.
- **/functions/cp.wddx.php**
  Funktionen zur (De-)Serialisierung von WDDX-Datenpaketen.
- **/functions/cp.xml.php**
  Funktionen zur Erstellung und Bearbeitung von XML-Dateien (z.B. RSS/RDF-Feeds).
- **/functions/cp.zip.php**
  Funktionen zur Erstellung und Bearbeitung von ZIP-Archiven.

Ein Großteil dieser PHP-Dateien wird in eigenen Unterkategorien ausführlicher behandelt. Die Einbindung sollte dabei immer mit *<?php include_once("functions/ [DATEINAME]"); ?>* vorgenommen werden, damit diese Dateien auch mehrfach innerhalb einer Seite eingebunden werden können.

6.1.2. cp_array_average()

Mit "cp_array_average()" lässt sich der Durchschnitt von Werten in einem nummerischen Array berechnen. Dies kann u.a. bei der Erstellung von Graphen zum Einsatz kommen. Der Funktion werden neben dem Array ein Start- und ein Endwert übergeben, wobei die beiden letzten Parameter optional sind.

Wenn der Startwert nicht angegeben ist, wird der erste Wert des Arrays genutzt. Bei einem fehlenden Endwert wird hingegen der letzte Wert des Arrays genutzt. Der Durchschnittswert kann erweitert werden, indem man die Differenz zwischen Start- und Endwert vergrößert.

```
$var = cp_array_average($array,"5","15"); // liefert den Durchschnittswert der Werte 5-15 des Arrays
```

6.1.3. cp_array_structure()

In **Contentpapst**-Modulen wird häufig mit komplexen Arrays gearbeitet, da diese eine sehr effiziente Abwicklung von Prozessen ermöglichen. Allerdings kann es passieren, dass man als Entwickler den Überblick über den Umfang eines Arrays verliert. Mit der Funktion "cp_array_structure()" gehört dieses Problem der Vergangenheit an.

"cp_array_structure()" gibt das übermittelte Array auf dem Bildschirm in einer beliebigen Tiefe aus. Die Funktion wird hierzu rekursiv durchlaufen. Beinhaltet da Array weitere Arrays, so werden diese ebenfalls durchlaufen.

```
cp_array_structure($test_array); // das Array "$test_array" wird angezeigt
```

6.1.4. cp_array_sum()

Mit "cp_array_sum()" lässt sich die Summe von Werten in einem numerischen Array berechnen. Dies kann u.a. bei der Erstellung von Statistiken zum Einsatz kommen. Der Funktion werden neben dem Array ein Start- und ein Endwert übergeben. Die beiden letzten Parameter sind optional.

Wenn der Startwert nicht angegeben ist, wird der erste Wert des Arrays genutzt. Bei einem fehlenden Endwert wird hingegen der letzte Wert des Arrays genutzt. Die Summe kann erweitert werden, indem man die Differenz zwischen Start- und Endwert erhöht.

```
$var = cp_array_sum($array,"5","15"); // das liefert die Summe der Werte 5-15 des Arrays
```

6.1.5. cp_currency_info()

Die Funktion "cp_currency_info()" gibt Auskunft über den Namen einer (Landes-)Währung, nachdem ihr ein dreistelliges Währungskürzel übermittelt wurde. Die Überprüfung ist dabei größtenteils konform zur ISO-Regelung "4217", erweitert um einige Industriestandards.

```
$var = cp_currency_info("EUR"); // sucht den Namen der Währung zum Kürzel "EUR"
```

6.1.6. cp_delete_cache()

Natürlich müssen im Cache abgelegte Inhalte auch vor Ablauf der festgelegten Cache-Zeit wieder gelöscht werden können, z.B. wenn eine Aktualisierung der Inhalte vorgenommen wurde. Hierzu muss der Funktion "cp_delete_cache()" der Namen der Cache-Datei übermittelt werden sowie der Ordner, in welchem die Datei abgelegt wurde. Standardmäßig ist dies der Ordner "cache".

```
cp_delete_cache("cache/", "test.tmp");
```

6.1.7. cp_get_cache()

Für umfangreichere Inhalte, welche nicht regelmäßig aktualisiert werden, empfiehlt es sich, diese in einem lokalen Cache zwischenzuspeichern. Die Funktion "cp_get_cache()" ermöglicht dabei den Abgleich der gewünschten Daten mit Inhalten im Cache. Der Funktion werden dazu als Parameter das Verzeichnis des Caches, der Dateiname und die maximale Cache-Zeit (in Sekunden) übergeben. Bei Erfolg wird TRUE zurückgegeben, andernfalls FALSE. Als Standard-Cache-Zeit werden 3600 Sekunden (d.h. 1 Stunde verwendet).

```
$var = cp_get_cache("cache/","test.tmp",3600);
```

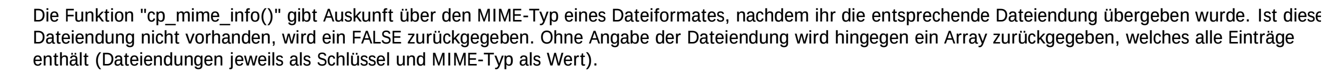
Prüft, ob die Datei "test.tmp" im Ordner "cache" vorhanden ist (diese darf nicht älter als 3600 Sekunden sein)

6.1.8. cp_lang_info()

Die Funktion "cp_lang_info()" gibt Auskunft über den Namen einer Sprache, nachdem ihr ein zweistelliges Namenskürzel übermittelt wurde. Die Überprüfung ist dabei konform zur ISO-Regelung "639-1".

```
$var = cp_lang_info("eo"); // sucht den Namen der Sprache zum Kürzel "eo" (Ergebnis ist "Esperanto")
```

6.1.9. cp_mime_info()



Die Funktion "cp_mime_info()" gibt Auskunft über den MIME-Typ eines Dateiformates, nachdem ihr die entsprechende Dateiendung übergeben wurde. Ist diese Dateiendung nicht vorhanden, wird ein FALSE zurückgegeben. Ohne Angabe der Dateiendung wird hingegen ein Array zurückgegeben, welches alle Einträge enthält (Dateiendungen jeweils als Schlüssel und MIME-Typ als Wert).

```
$var = cp_mime_info("mp3"); // gibt den MIME-Typ für "mp3" zurück (Ergebnis ist "audio/mpeg")
```

6.1.10. cp_module_exists()

Oftmals soll zwischen verschiedenen **Contentpapst**-Modulen eine Verbindung hergestellt werden. Da jedoch nicht immer gewährleistet werden kann, dass diese Module auch vorhanden sind, muss vorab ihre Verfügbarkeit überprüft werden. Hierfür kann die Funktion "cp_module_exists()" genutzt werden. Der Funktion wird der Modul-Name (d.h. der Namen des Ordners, in dem das Modul liegt) übergeben. Als Ergebnis werden TRUE oder FALSE zurückgeliefert.

```
$var = cp_module_exists("forum"); // prüft, ob das Foren-Modul vorhanden ist
```

6.1.11. cp_random_string()

"cp_random_string()" ist eine Funktion zur Erzeugung von zufallsgenerierten Werten. Diese bestehen aus einer Kombination von Buchstaben und Zahlen und können beispielsweise bei der Generierung von Passwörtern zum Einsatz kommen. Über den Parameter "lenght" kann die gewünschte Länge vorgegeben werden.

```
$var = cp_random_string("10"); // liefert einen 10-stelligen Wert zurück (z.B. "A94ZUAIQ0P")
```

6.1.12. cp_read_cache()

Nach der Abfrage mittels "cp_get_cache()", ob eine Seite im Cache vorhanden ist, kann mit "cp_read_cache()" bei Erfolg diese Seite aus dem Cache geladen und angezeigt werden.

```
$var = cp_read_cache("cache/","$full_name"); // liest den Cache von "$full_name" aus dem Ordner "files"
```

6.1.13. cp_set_cache()

Für umfangreichere Inhalte, welche nicht regelmäßig aktualisiert werden, empfiehlt es sich, diese in einem lokalen Cache zwischenzuspeichern. "cp_set_cache ()" dient der Speicherung von Inhalten im Cache. Der Funktion wird dabei neben dem Inhalt noch ein Ablageort mitgeteilt. Dieser setzt sich aus dem Verzeichnis (standardmäßig ist dies der Ordner "cache") und dem Dateinamen zusammen. Letzerer sollte optimal aus einer md5-Verschlüsselung des originalen Dateinamens samt Endung ".tmp" bestehen.

```
$var = cp_set_cache($data, "cache/", $output); // speichert "$data" unter dem Namen "$output" im Ordner "cache"
```

6.1.14. cp_validate_color()

Mit der Funktion "cp_validate_color()" lässt sich die Gültigkeit eines Farbwertes im HTML-Schema ermitteln. Ein Farbwert muss dabei mit einer Raute "#" beginnen und zusammen mit dieser Raute insgesamt 7 Zeichen lang sein. Wenn der Farbwert gültig ist, so wird TRUE zurückgegeben, andernfalls ein FALSE.

```
$var = cp_validate_color("#604FFFF"); // testet, ob der Farbwert gültig ist (Ergebnis: FALSE, da zu lang)
```

6.1.15. cp_validate_email()

Überprüfen Sie Email-Adressen auf ihre Gültigkeit mit der Funktion "cp_validate_email()". Die gewünschte Email-Adresse wird der Funktion als Parameter übergeben und dort anhand eines regulären Ausdrucks überprüft. Es wird nur die Struktur überprüft, nicht ob diese Email-Adresse auch genutzt wird. Ist die Email-Adresse korrekt, so wird ein TRUE zurückgegeben, anderfalls ein FALSE.

```
$var = cp_validate_email("contact@sandoba.de"); // testet auf die Email-Adresse "contact@sandoba.de"
```

6.1.16. cp_validate_number()

Mit "cp_validate_number()" überprüfen Sie, ob Inhalte andere Zeichen als die Zahlen 0-9 enthalten. Ist dies der Fall, so wird FALSE zurückgegeben, anderfalls TRUE.

```
$var = cp_validate_number("19. August 2003"); // überprüft den übergebenen Inhalt (Ergebnis: "FALSE")
```

6.1.17. cp_validate_string()

Wenn Sie ein Mitgliedersystem oder ähnliches nutzen und hierfür per Formular übermittelte Inhalte überprüfen müssen, können Sie die Funktion "cp_validate_string()" benutzen. Diese erlaubt nur alphanummerische Zeichen. Der Funktion kann außer dem String noch eine minimale bzw. maximale Länge übergeben werden.

```
$var = cp_validate_string($string,"5","20"); // "$string" muss min. 5 und darf max. 20 Zeichen lang sein
```

6.1.18. cp_word_shorten()

Gerade in Foren, Gästebüchern oder anderen Modulen, welche die Interaktion mit dem Besucher einbeziehen kann es oftmals zu Problemen kommen, wenn beispielsweise das Layout der Seite durch zu lange Wörter aufgesprengt wird. Dies lässt sich mit der Funktion "cp_word_shorten()" hervorragend verhindern.

Übermitteln Sie dieser Funktion einfach ein Wort bzw. einen Text und geben Sie als zweiten Parameter die maximale Länge eines Wortes an. Standardmäßig sind dies 50 Zeichen. Längere Wörter werden auf (n-3) Zeichen verkürzt. Diesen wird anschließend "..." angehängt.

```
$var = cp_word_shorten("dies_ist_nur_ein_simpler_test","20"); // verkürzt den Text auf "dies_ist_nur_ein_..."
```

6.2. Allgemeines zur Modulentwicklung

*Diese Informationen beschreiben die Entwicklung neuer Module für **Contentpapst** auf Basis des neuen objektorientieren Ansatzes. **Contentpapst** verwendet hierbei Klassen und deren Vorteile wie Ableitung, Vererbung und Wiederverwendbarkeit. Sie sollten sich also gut mit Klassen auskennen, um damit arbeiten zu können.*

6.2.1. Grundsätzlicher Aufbau

Jedes Modul in Contentpapst, welches den objektorientierten Ansatz verwendet, besteht aus mindestens drei Klassen, die in jeweils eigenen Dateien gespeichert werden:

- **class.base.php**
  Beherbergt die Klasse "contentpapst_[MODULNAME]". Diese stellt grundlegende Methoden zum Objekthandling bereit und wird daher u.a. von den anderen Klassen des Modules aber auch extern genutzt.
- **class.admin.php**
  Die Klasse "contentpapst_gui_[MODULNAME]" generiert den Administrationsbereich des Modules mit allen Formularen, Menüs usw.
- **class.index.php**
  Diese Klasse heißt ebenfalls "contentpapst_gui_[MODULNAME]". Jedoch wird diesmal das Frontend, d.h. der öffentlich zugängliche Bereich generiert.

Wenn Sie sich jetzt testweise einige derart aufgebaute Module anschauen, werden Sie sicherlich feststellen, dass sich die Klassennamen jeweils leicht unterscheiden. Denn zur besseren Unterscheidung muss "[MODULNAME]" immer durch den Verzeichnisnamen ersetzt werden, in dem das Modul liegt. Die Basis-Klasse des Modules "languages" heißt also dementsprechend "contentpapst_languages".

Zu beachten ist auch, dass die Klassen in den Dateien "class.admin.php" und "class.index.php" von der Klasse "contentpapst_[MODULNAME]" abgeleitet werden müssen (über "... extends contentpapst_[MODULNAME]"), damit deren Methoden genutzt werden können.

Benötigt werden weiterhin die folgenden Dateien / Ordner für jedes Modul:

- **config.php**
  Zentrale Konfigurationsdatei für das Modul mit wesentlichen Einstellungen (u.a. Installation notwendig (j/n), Versionsnummer, Name und Informationen zum Autor).
- **class.webservices.php**
  Die Klasse "contentpapst_webservices_[MODULNAME]" ermöglicht den Zugriff auf die Daten des Modules über XML-Webservices.
- **index.html**
  Leere HTML-Datei um den direkten Aufruf des Ordners im Browser zu verhindern.
- **languages / [SPRACHE] / index.php**
  Für jede im System vorhandene Sprache muss eine entsprechende Sprachdatei in den Unterordnern vorhanden sein.
- **Ordner "images" und "templates"**
  In diesen beiden Ordnern sollten die vom Modul genutzten Grafiken und Vorlagen untergebracht werden.

6.2.2. Datei "config.php": Die Konfigurationsdatei

In der zentralen Konfigurationsdatei für jedes Modul werden wichtige Einstellungen und Informationen festgehalten. Die Datei muss als "config.php" im Hauptordner des Modules abgelegt werden.

```
<?php
$module_config["example"]["name"] = "example"; // Interner Name des Modules (wie der Ordnername)
$module_config["example"]["version"] = "1.0"; // Versionsnummer des Modules
$module_config["example"]["date"] = "01.05.2004"; // Datum der letzten Arbeiten an diesem Modul
$module_config["example"]["author"] = "sandoba.de medien agentur"; // Name des Modul-Entwicklers
$module_config["example"]["email"] = "contact@sandoba.de"; // Email-Adresse des Modul-Entwicklers
$module_config["example"]["website"] = "http://www.sandoba.de"; // Webseite des Modul-Entwicklers
$module_config["example"]["image"] = ""; // URL zu einem Bild des Modules (für die Übersicht im Administrationsbereich)
$module_config["example"]["status"] = "1"; // Status des Modules (1 für Aktiviert, 0 für Deaktiviert, -1 für Deaktiviert in Admin- und Benutzer-Bereich)
```

```
$module_config["example"]["install"] = "0"; // Bei "install" = 1 muss das Modul erst noch per Admin-Menü installiert werden
$module_config["example"]["show_list"] = "0"; // Im Administrationsbereich als Modul sichtbar (1 für Aktiviert, 0 für Deaktiviert)
?>
```

6.2.3. Datei "class.admin.php": Der Administrationsbereich

Der Administrationsbereich eines Modules wird über die Datei "class.admin.php" erstellt. Auch dies ist eine Klasse (genauere Informationen auch unter "Aufbau des Konstruktors").

```
<?php
class contentpapst_gui_example extends contentpapst_example {

    var $module_configuration;
    var $module_language;

    /*
        class contentpapst_gui_example::contentpapst_gui_example
        Konstruktur, dient der Initialisierung der Klasse und einiger Variablen.
    */

    function contentpapst_gui_example() { }

    /*
        class contentpapst_gui_example::install
        SQL- und Dateianweisungen zur Installation des Modules.
    */

    function install() { }

    /*
        class contentpapst_gui_example::uninstall
        SQL- und Dateianweisungen zur Deinstallation des Modules.
    */

    function uninstall() { }

}
?>
```

6.2.4. Datei "class.base.php": Basis der Modulentwicklung

Die Aufteilung in verschiedene Klassen bietet die Möglichkeit, einzelne Methoden in anderen Bereichen wiederzuverwenden, ohne dass diese erneut in einer separaten Datei gespeichert werden müssten. Dies spart Zeit, da Änderungen nur einmal durchgeführt werden müssen. Zudem erhöht es die Übersichtlichkeit, da die Basisklasse für jedes Modul in der Datei "class.base.php" gespeichert ist. Wenn Sie eine neue Klasse erstellen, die auf die Methoden der Basisklasse eines Modules zurückgreifen soll, dann müssen Sie die Datei "class.base.php" per "include_once()" einbinden und die neue Klasse von der Basisklasse ableiten ("... extends contentpapst_[MODULNAME]").

```
<?php
class contentpapst_example { }
?>
```

6.2.5. Datei "class.index.php": Der öffentliche Bereich

Der öffentliche Bereich eines Modules wird über die Datei "class.index.php" erstellt. Auch dies ist eine Klasse (genauere Informationen auch unter "Aufbau des Konstruktors").

```
<?php
class contentpapst_gui_example extends contentpapst_example {

    var $module_configuration;
    var $module_language;

    /*
        class contentpapst_gui_example::contentpapst_gui_example
        Konstruktur, dient der Initialisierung der Klasse und einiger Variablen.
    */

    function contentpapst_gui_example() {

        GLOBAL $file, $lang, $mode;

        $this->module_language = module_language($file) + $lang;

        $this->module_configuration["functions"] = array(
                                                "main"=>array("do_main", array("show"))
                                            );

    }
```

```
    /*
        class contentpapst_gui_example::metatags
        Behandelt die Einblendung der Metatags.
    */

    function metatags() {}

    /*
        class contentpapst_gui_example::search
        Stellt die Verbindung zur Volltextsuche her.
    */

    function search() {}

    ##################################################################

    /*
        class contentpapst_gui_example::do_main
    */

    function do_main($show) {

        // Einzelne Methoden...

    }

}
?>
```

6.2.6. Datei "class.webservices.php": Zugriff auf Inhalte über Webservices

Über die Datei "class.webservices.php" können Sie das Modul um eine Webservices-Schnittstelle erweitern. D.h. Inhalte können als XML exportiert und in andere Systeme importiert bzw. beliebig dargestellt werden. Hier sehen Sie die Basis-Datei, die dann je nach Modul erweitert werden muss:

```
<?php
class contentpapst_webservices_example {

    /*
        class contentpapst_webservices_example::process
        Methode, dient der Verarbeitung der ankommenden Informationen.
    */

    function process($command, $form) {

        include("config.php");

        include_once("functions/cp.string.php");
        include_once("functions/cp.url.php");
        include_once("functions/cp.xml.php");

        GLOBAL $xml, $file, $sql;

        $content = "";

        $xml->set_content("example/error", "Sorry, there are no exact matches for your request.");

        $content = $xml->show("xml");

        RETURN $content;

    }

}
?>
```

6.2.7. Aufbau des Konstruktors

Der Konstruktor initialisiert einige grundlegende Variablen der Klasse, welche den Aufbau des Modules genauer beschreiben. U.a. erfüllt der Konstruktor folgende Aufgaben:

- gibt Titel und Beschreibung der Klasse vor
- liest die Sprachdatei für das Modul ein
- sorgt für den Aufbau des Navigationsmenüs (obere Leiste und Seitentitel)
- liefert Informationen über die einzelnen Funktionen und deren Parameter
- teilt dem System mit, welche Funktionen welcher Rechte bedürfen

Jeder dieser Aufgaben werden wir uns nachfolgend genauer widmen.

```
$this->module_language = module_language($file) + $lang;
```

Mit dieser Zeile wird die Sprachdatei für das Modul eingelesen. Das vorhandene Array $lang wird anschließend daran angehängt, damit in der Klasse auf alle Sprachelemente zugegriffen werden kann.

```
$this->module_configuration["basics"] = array(
    "title"=>$this->module_language["$file"]["module_title"],
    "description"=>$this->module_language["$file"]["module_desc"]
);
```

Zuerst wird dem System der Titel und die Beschreibung des Modules mitgeteilt. Bereits diese Sprachelemente stammen aus der zuvor eingelesenen Sprachdatei. U.a. werden die Informationen zur Darstellung des Modules im Administrationsbereich benötigt.

```
$this->module_configuration["navigation"] = array(
    "main"=>array("Übersicht", "0", array()),
    "add_article"=>array("Artikel hinzufügen", "0", array()),
    "search"=>array("Suche", "0", array("seach_title"=>"Nach Titel", "search_keyword"=>"Nach Schlüsselwort"))
);
```

Das Array für die Erstellung der Navigation ist schon ein wenig komplexer. Dieses Array steuert nicht nur die obere Navigationsleiste, sondern - sofern eine bestimmte Funktion/Methode aufgerufen wurde - auch die Anzeige direkt darunter (den Titel der Seite). Zur besseren Verdeutlichung wurden in den folgenden Beispielen die Sprachelemente direkt integriert.

Im Wesentlichen wird hier jeder Methode ein Eintrag zugeordnet. Die Methode "add_article" erhält den Titel "Artikel hinzufügen". Die darauffolgende Zahl (im Beispiel eine "0") bestimmt, ob der Eintrag normal (keine Fettschrift, "0"), als Hauptmenüpunkt ("1") oder gar nicht angezeigt wird ("2"). Am Ende folgt noch ein Array. Dieses ist nur notwendig, wenn für den Eintrag "1" ausgewählt wurde. Dann werden in diesem Array weitere Unterpunkte eingetragen (siehe die Methode "search" mit der Zuweisung von Schlüssel und Inhalt für die Unterpunkte).

Der Schlüssel "main" steht für die Übersichts-Seite, d.h. die Seite, welche beim Aufruf des Modules im Administrations-Menü als erstes erscheint.

```
$this->module_configuration["functions"] = array(
    "main"=>array("do_main", array()),
    "search"=>array("do_search", array()),
    "search_title"=>array("do_search_title", array("form")),
    "search_keyword"=>array("do_search_keyword", array("keyword"))
);
```

Im nächsten Schritt wird festgelegt, welche Funktionen/Methoden für das Modul innerhalb der Klasse existieren. Hierzu wird ähnlich dem Array für die Navigation ein weiteres Array mit den gleichen Schlüsseln aufbaut. Allerdings wird nun jedem Schlüssel der Name einer Methode zugewiesen. Direkt anschließend folgt ein Array mit den Parametern, welche beim Aufruf der Methode benötigt werden. Dies ermöglicht es Contentpapst, gezielt die gewünschten Inhalte als Parameter zu übergeben, ohne Sicherheitslöcher durch globale Variablen etc. zu öffnen.

```
$this->module_configuration["rights"] = array(
    "all"=>array("languages",8),
    "do_main"=>array()
);
```

Direkt für die Sicherheit verantwortlich ist das obenstehende Array. In diesem wird festgelegt, welche Methode welche Rechte / Gruppenzugehörigkeit erfordert. Den Methoden kann der Name des Rechtes (über die Rechteverwaltung zu ermitteln) zugewiesen werden, welches vorhanden sein muss. Optional ist die Angabe einer Gruppen-ID. Alle Mitglieder die sowohl das Recht UND eine Gruppen-ID größer oder gleich der erforderlichen ID besitzen, können die Methode aufrufen. Allen anderen wird eine Fehlermeldung ausgegeben. Der Schlüssel "all" gibt in diesem Fall an, welche Erfordernisse für alle Methoden gelten.

6.2.8. Struktur der Datenbanktabellen

In dieser Auflistung finden Sie alle durch Contentpapst in der Grundversion (d.h. ohne zusätzliche Module wie CP::Shop) genutzten Datenbanktabellen. In diesen werden die kompletten Inhalte gespeichert:

- **_adminfaq* - Modul "Anleitung", Einträge aus der Anleitung
- **_adminfaqsec* - Modul "Anleitung", Kategorien
- **_article* - Modul "Artikel", Speicherung der Artikel
- **_article_views* - Modul "Artikel", Sichtbarkeit der Artikel für Gruppen
- **_articlerating* - Modul "Artikel", Bewertung der Artikel
- **_articlesection* - Modul "Artikel", Kategorien
- **_banner* - Modul "Werbung", Speicherung der einzelnen Werbeelemente
- **_bannerclients* - Modul "Werbung", Speicherung der Werbekunden
- **_blocked* - Zugriffssperre für Einträge einzelner Module
- **_bookmarks* - Persönliche Bookmarks der Benutzer
- **_category* - Modul "News", Kategorien
- **_comment* - Modul "News", Kommentare zu den News
- **_domains* - Datenbanktabelle mit Domainkürzeln und Ländernamen
- **_forms* - CP::Forms, Allg. Angaben zu den Modulen
- **_forms_access* - CP::Forms, Zugriffsrechte für die Module
- **_forms_collections* - CP::Forms, Speicherung der Collections
- **_forms_datasets* - CP::Forms - Allg. Angaben zu den Datensätzen der Module
- **_forms_fields* - CP::Forms - Felder der einzelnen Module
- **_forms_fields_access* - CP::Forms - Zugriffsrechte für die Felder
- **_forms_values* - CP::Forms, Werte der einzelnen Datensätze

- *_gallery* - Modul "Galerie", Speicherung der Bilddaten
- *_gallery_cat* - Modul "Galerie", Kategorien
- *_gallery_error* - Modul "Galerie", Speicherung der gemeldeten Fehler
- *_gallery_rating* - Modul "Galerie", Bewertungen der Bilder
- *_guestbook* - Modul "Gästebuch", Speicherung der Einträge
- *_links* - Modul "Links", Speicherung der Einträge
- *_links_cat* - Modul "Links", Kategorien
- *_links_comment* - Modul "Links", Kommentare zu den Links
- *_links_rating* - Modul "Links", Bewertungen der Links
- *_log* - Zentrale Logtabelle über Benutzeraktionen
- *_navigation* - Modul "Navigation", Struktur der Navigationen
- *_news* - Modul "News", Speicherung der News
- *_newsletter* - Modul "Newsletter", Versandte Newsletter
- *_newsletterfilter* - Modul "Newsletter", Filter-Einstellungen
- *_newslettersections* - Modul "Newsletter", Eingetragene Newsletter
- *_newsletteruser* - Modul "Newsletter", Newsletter-Abonnenten
- *_notes* - Persönliche Notizen für den Administrationsbereich
- *_options* - Konfigurationstabelle mit allen Einstellungen
- *_permalinks* - Permalinks für verschiedene Inhalte, wie z.B. Artikel, News und Shop-Artikel
- *_polls_answer* - Modul "Umfragen", Speicherung der Stimmen
- *_polls_question* - Modul "Umfragen", Fragegestellungen
- *_random* - Zufallsgesteuerte Elemente der Webseite
- *_ratings* - Bewertungen für einzelne Elemente der Webseite
- *_sites* - Angaben zu den einzelnen Mandanten des Systems
- *_stats_misc* - Modul "Statistik", Logging von Suchabfragen etc.
- *_stats_power* - Modul "Statistik", Speicherung der Benutzerzugriffe
- *_tags* - Tags (Schlüsselwörter) für verschiedene Inhalte der Webseite
- *_tasks* - Modul "Aufgaben",
- *_tasks_comments* - Modul "Aufgaben", Kommentare zu den Aufgaben
- *_tasks_sections* - Modul "Aufgaben", Kategorien
- *_tracking* - Modul "Newsletter", Tracking von Links in den Newslettern
- *_user* - Benutzerverwaltung, Speicherung der Benutzer
- *_user_grouprights* - Benutzerverwaltung, Rechtvergabe für einzelne Module
- *_user_groups* - Benutzerverwaltung, Speicherung der Gruppen
- *_user_info* - Benutzerverwaltung, Speicherung erweiterter Informationen zu Benutzern
- *_user_rights* - Benutzerverwaltung, Allgemein Rechte für die Module
- *_webservices* - Modul "Webservices", Speicherung der Angaben zu den RDF/RSS-Feeds

6.3. Benutzer

Die Datei "/functions/cp.user.php" stellt Funktionen rund um die Benutzer/Mitglieder des Systemes zur Verfügung. Dies schließt u.a. Funktionen zur Ermittlung von IP-Adressen und zur Validierung der Zugriffsrechte mit ein. Um diese Funktionen nutzen zu können, muss die Datei "/functions/cp.user.php" per "include()" oder "include_once()" integriert werden.

6.3.1. cp_get_userip()

Die Funktion "cp_get_userip()" ermittelt die IP-Nummer des aktuellen Besuchers und kann u.a. bei der Speicherung von Beiträgen oder bei Abstimmungen (Umfrage-Modul) genutzt werden.

6.3.2. cp_get_username()

"cp_get_username()" liefert einen Benutzernamen zurück. Dazu wird der Funktion die Nummer des jeweiligen Benutzers übergeben. Ist kein Benutzer mit dieser Nummer vorhanden, dann wird FALSE zurückgegeben.

6.3.3. cp_user_info()

Mit der Funktion "cp_user_info()" lassen sich gezielt anhand der Nummer eines Benutzers weitere Informationen über diesen Benutzer ermitteln. Der Funktion wird zuerst die Nummer übergeben, als zweiter Parameter folgt der Name des gewünschten Feldes (z.B. "username", "street", "email"). Die Funktion liefert den Inhalt dieses Feldes dann zurück. Ist kein Benutzer mit dieser Nummer vorhanden, dann wird FALSE zurückgegeben.

6.3.4. cp_user_lastlogin()

Mit dem integrierten Logging-System ist es möglich, gezielt bestimmte Aktionen eines Benutzers aus der Vergangenheit herauszugreifen und diese darzustellen. Die Funktion "cp_user_lastlogin()" liefert für den aktuellen Benutzer das Datum des letzten Logins zurück. Standardmäßig ist der Rückgabewert im Format "d.m.Y", über einen optionalen Parameter kann dies jedoch geändert werden. So liefert "cp_user_lastlogin("d.m.y - H:i")" beispielsweise Datum und Uhrzeit des letzten Logins.

6.3.5. cp_user_rights()

Diese Funktion ist besonders wichtig für den Zugriff von Benutzern auf geschützte Bereiche. Der Funktion wird der interne Name eines Rechtes aus der Rechteverwaltung übergeben. Anschließend wird überprüft, ob die Gruppe des aktuellen Benutzers auf diesen Bereich Zugriff hat. Ist dies der Fall, wird TRUE zurückgegeben, andernfalls FALSE.

6.3.6. cp_user_grouprights()

Diese Funktion ist eine Erweiterung der Funktion "cp_user_rights()". Über "cp_user_grouprights()" können die detaillierten Benutzerberechtigungen eines Benutzers für den Datensatz eines einzelnen Modules abgefragt werden. Dies wird

beispielsweise bei den Kategorien des Artikel-Modules für die Vergabe von Berechtigungen zum Hinzufügen/Bearbeiten/Löschen von Artikeln genutzt. Der Funktion muss der Modulname, die Nummer des Datensatzes und die Nummer der Berechtigung übergeben werden. Rückgabewerte sind je nach Ergebnis "0" (false) oder "1" (true).

6.4. CSS

Mit den CSS-Funktionen können Sie vorhandene CSS-Dateien auslesen, bearbeiten und wieder speichern. Auch das Anlegen neuer CSS-Dateien ist möglich. Um diese Funktionen in Module einzubinden, muss die Datei "/functions/cp.css.php" per "include()" oder "include_once()" eingebunden werden.

6.4.1. cp_css_input()

Eine CSS-Datei kann mit der Funktion "cp_css_input()" eingelesen und in ein Array umgewandelt werden. Dazu muss der Funktion lediglich der Pfad zur Datei übergeben werden.

```
$var = cp_css_input("stylesheet.css"); // Datei "stylesheet.css einlesen
```

6.4.2. cp_css_output()

Die Funktion "cp_css_output()" verwandelt ein zweidimensionales Array in eine fertige CSS-Datei. Das Array kann aus einer durch "cp_css_input()" eingelesenen Datei stammen bzw. manuell erstellt werden. Neben dem Array muss der Funktion der Pfad zur neuen Datei übergeben werden.

```
cp_css_output("style.css",$array); // erstellt die Datei "style.css" mit den Daten aus "$array"
```

6.5. Dateifunktionen

Mit den Methoden dieser Klasse können verschiedenste Dateioperationen durchgeführt werden. Beispielsweise können Dateien und komplette Ordner gelöscht werden, es können Auflistungen der Dateien in einem Ordner angefordert werden usw. Um diese Methoden in Module einzubinden, muss die Datei "/functions/cp.files.php" per "include()" oder "include_once()" eingebunden werden.

6.5.1. Funktionsübersicht dieser Datei

In der Klasse "contentpapst_files" in der Datei "/functions/cp.files.php" sind u.a. die folgenden Funktionen/Methoden für die Erstellung und Bearbeitung von Dateien und Ordnern vorhanden:

extension	Liefert die Dateierweiterung einer Datei zurück (z.B. "exe").
list_dirs	Liefert ein Array mit den Unterordnern in einem Ordner zurück.
list_files	Liefert ein Array mit den Dateien in einem Ordner zurück.
delete_dirs	Löscht ein Array an Ordnern aus dem Dateisystem.
delete_files	Löscht ein Array an Dateien aus dem Dateisystem.

6.6. Datumsfunktionen

Mit diesen Methoden der Date-Klasse ist das Berechnen beliebiger Datumsangaben möglich. Ausgehend von den übergebenen Parametern können Schaltjahre, Wochentage usw. genau ermittelt werden. Um die Methoden in Module einzubinden, muss die Datei "/functions/cp.date.php" per "include()" oder "include_once()" eingebunden werden.

6.6.1. check_date()

Alternativ zur Methode "is_valid_date()" kann zur Überprüfung eines Datums die Methode "check_date()" eingesetzt werden. Diese nutzt intern die Funktion "checkdate()" von PHP. Für die Überprüfung wird der Methode ein Datum im Format TT.MM.YY bzw. TT.MM.YYYY übergeben. Mögliche Rückgabewerte sind "1" (true) und "0" (false).

```
$date->check_date("30.07.2005"); // liefert true zurück für den 30.07.2005
```

6.6.2. days_per_month()

"days_per_month()" wird auch von anderen Datumsfunktionen eingesetzt und ermittelt die Anzahl der Tage in einem Monat in einem bestimmten Jahr. Dabei werden auch Schaltjahre mit beachtet. Der Rückgabewert ist die entsprechende Anzahl an Tagen.

```
$var = $date->days_per_month("4","2007"); // Anzahl der Tage im Monat 04/2007 (Ergebnis: "30")
```

6.6.3. first_monday()

Die Methode "first_monday()" ermittelt den Montag der ersten Kalenderwoche eines Jahres. Dieser muss allerdings nicht immer innerhalb des Jahres, d.h. im Januar, sondern kann auch im Dezember liegen (z.B. am 29.12. für das Jahr 2004). Die Methode erhält als einzigen Parameter ein Jahr übergeben (4-stellig). Wird dieser Parameter nicht übergeben, dann wird das aktuelle Jahr verwendet. Der Rückgabewert ist das Datum des ersten Montags des Jahres im Timestamp-Format.

```
$date->first_monday("2005"); // liefert für "2005" den Timestamp "1104274800" zurück
```

6.6.4. is_leap_year()

"is_leap_year()" prüft, ob das übergebene Jahr ein Schaltjahr ist. Wird der Methode kein Jahr übergeben, so wird das aktuelle Jahr als Parameter genommen. Bei Erfolg gibt die Methode "true" zurück, andernfalls "false".

```
$var = $date->is_leap_year("2009"); // prüft das Jahr 2009 (Ergebnis: "false")
```

6.6.5. is_valid_date()

Mit dieser Methode kann geprüft werden, ob ein Datum auch wirklich existiert. D.h. Tag, Monat und Jahr müssen vorhanden sein und bestimmte Begrenzungen einhalten (Tage pro Monat, Schaltjahr usw.). Der Methode "cp_valid_date()" werden hierzu Tag, Monat und Jahr übergeben. Als Ergebnis wird "true" oder "false" zurückgegeben.

```
$var = $date->is_valid_date("29","3","2003"); // prüft den 29.03.2003 (Ergebnis: "true")
```

6.6.6. mysql_timestamp()

Diese Methode benötigt keinen Parameter, sondern liefert einfach nur die aktuelle Zeit bzw. das aktuelle Datum im von MySQL benötigten Timestamp-Format zurück.

```
$var = $date->mysql_timestamp(); // liefert das aktuelle Datum als MySQL-Timestamp
```

6.6.7. week_monday()

Als Erweiterung der Methode "first_monday()" wird "week_monday()" eingesetzt. Dieser Methode werden eine Kalenderwoche (1-52) sowie optional ein Jahr (4-stellig) übergeben. Als Rückgabewert erhalten Sie im Timestamp-Format das Datum des Montags in dieser Kalenderwoche.

```
$date->week_monday("3", "2005"); // liefert den Timestamp für den Montag der 3. Kalenderwoche 2005
```

6.7. Formulare

*Alle Formulare in **Contentpapst** werden über die API erzeugt, da sich hier schnell Überprüfungen der Formulardaten, Veränderungen der Felder usw. einbauen lassen. Die Formular-Funktionen stehen direkt in jeder Datei zur Verfügung, welche als Modul integriert ist. Eine Einbindung weiterer Dateien ist nicht notwendig.*

6.7.1. form_start_wysiwyg()

Wird in einem Formular ein WYSIWYG-Editor eingesetzt, so muss dieses Formular mit der Funktion "form_start_wysiwyg()" begonnen werden. Der Funktion wird hierzu beim Aufruf ein Formular-Ziel und die gewünschte Übergabemethode ("get"/"post") übergeben. Standardmäßig wird "post" genutzt. "form_start_wysiwyg()" nimmt zusätzlich eine Überprüfung des Browsers des Benutzers vor.

```
form_start_wysiwyg("admin.php?file=controlling&mode=add_info","post");
// Formular wird per "post" an das Modul "controlling" gesendet
```

6.7.2. form_wysiwyg()

Die Funktion "form_wysiwyg()" ist das Kernstück des in Contentpapst integrierten WYSIWYG-Editors. Mit dieser Funktion lässt sich der Editor komplett frei strukturieren. Alle Optionen des Editors lassen sich einzeln abschalten, so dass sowohl eine eingeschränkte Version mit den wichtigsten Optionen zur Textbearbeitung sowie auch ein sehr komplexer WYSIWYG-Editor für die eigenen Module realisiert werden können.

Die Funktion benötigt zur Integration des WYSIWYG-Editors alle Informationen in Form eines Arrays (siehe Beispiel unten). Zusätzlich muss das Formular mit der Funktion "form_start_wysiwyg" statt wie üblich über "form_start" begonnen werden.

url_rewrite	Bearbeitungsmodus ("admin" oder "index")
width	Breite des Eingabefeldes
height	Höhe des Eingabefeldes
title	Name des Editors
type	Typ (1 = neuer Inhalt, 2 = Datei öffnen [dann ist "content" der Dateiname])
content	Vorgegebener Inhalt des Eingabefeldes
hide_buttons	Einfaches Textfeld ohne Buttons
width_percent	Breite ist in Prozent angegeben
height_percent	Höhe ist in Prozent angegeben
element / start	Beginn einer Buttonleiste
element / stop	Ende einer Buttonleiste
elements / 0 … n	Buttonleiste mit Array der Buttons (siehe unten)
allow	Aktivierte Buttons ("all" = Alle Buttons sichtbar)

selected_border_color	Rahmenfarbe bei Auswahl eines Buttons
selected_background	Hintergrundfarbe bei Auswahl eines Buttons
hover_border_color	Rahmenfarbe bei Mouseover eines Buttons
hover_background	Hintergrundfarbe bei Mouseover eines Buttons
unselected_border_color	Rahmenfarbe bei Abwahl eines Buttons
unselected_background	Hintergrundfarbe bei Abwahl eines Buttons
return	Ausgabemodus (0 = direkt, 1 = Rückgabe an Variable)

Das untenstehende Beispiel zeigt den WYSIWYG-Editor wie er auch im Artikel-Modul zum Einsatz kommt. Hier werden u.a. verschiedene Farben für die einzelne Buttons vergeben, zwei Leisten mit den gewünschten Buttons erzeugt und eine Größe für das Eingabefeld vergeben.

```
$options["url_rewrite"] = "admin";
$options["width"] = "618";
$options["height"] = "350";
$options["title"] = "articledesc";
$options["type"] = "1";
$options["content"] = "";
$options["hide_buttons"] = "0";
$options["width_percent"] = "0";
$options["height_percent"] = "0";

$options["row"]["start"] = "<div style="border: 0px; border-color: #000000; border-style: solid">";
$options["row"]["stop"] = "</div>";

$options["elements"]["0"] = array("cut", "copy", "paste", "separator", "print", "saveas", "separator", "undo", "redo",
"separator", "bold", "italic", "underline", "separator", "justifyleft", "justifycenter", "justifyright", "justifyfull", "separator",
"insertorderedlist", "insertunorderedlist", "separator", "forecol", "backcol");

$options["elements"]["1"] = array("font", "size", "images", "special", "separator", "removeformat", "word", "separator",
"inserthorizontalrule", "outdent", "indent", "separator", "toggleview", "preview", "separator", "createlink", "unlink",
"enlarge", "table");

$options["allow"] = array("all");

$options["selected_border_color"] = "#E1EAF7";
$options["selected_background"] = $background;
$options["unselected_border_color"] = $tableback;
$options["hover_background"] = $background;
$options["hover_border_color"] = $tableback;

form_wysiwyg($options);
```

Folgende Buttons sind vorhanden für die Integration in den Buttonleisten. Die Verfügbarkeit eines Buttons ist davon abhängig, ob der CP::Editor für erweiterte Funktionen integriert ist.

Übersicht der Buttons für die Buttonleisten	
cut / copy / paste / print / saveas	Ausschneiden, Kopieren, Einfügen, Drucken, Speichern
undo / redo	Rückgängig/Wiederherstellen
bold / italic / underline / strikethrough	Fett, Kursiv, Unterstrichen, Durchgestrichen
justifyleft / justifycenter / justifyright / justifyfull	Ausrichtung links, zentriert, rechts und Blocksatz
insertorderedlist / insertunorderedlist	Nummerierung/Aufzählung
forecol / backcol	Text- und Hintergrundfarbe
removeformat	Formatierung entfernen
outdent / indent	Einzug verkleinern/vergrößern
images / special	Grafik und Sonderzeichen einfügen
word	Word-Quelltext aufräumen
preview	Vorschau anzeigen
createlink / unlink	Link einfügen/entfernen
inserthorizontalrule	Horizontale Linie einfügen
toggleview	Wechsel zwischen WYSIWYG- und Text-Modus
selectall	Alles auswählen
unselect	Alles abwählen
toupper / tolower	Großschrift/Kleinschrift
superscript / subscript	Schrift hochgestellt/tiefgestellt
table	Tabelle einfügen
br	Einfacher Zeilenumbruch
pre	Vorformatierter Text (Quellcode)
email	Formular für Email-Adresse
link	Erweiterte Linkeinstellungen
button	Schaltfläche
checkbox	Kontrollkästchen
radio	Optionsfeld
textarea	Textbereich
textfield	Textfeld
form	Formular

hidden	Verstecktes Feld
dropdown	Dropdownfeld
find	Suchen
anchor	Textmarke
zoom	Auswahlmenü für Zoomstufe
float_left	Inhalte ausrichten (links)
float_right	Inhalte ausrichten (rechts)

6.8. FTP - File Transfer Protocol

Mit den Methoden dieser Klasse können Sie per PHP mit FTP-Servern interagieren, Datei verschieben, hochladen und entfernen usw. Auch das Setzen von Zugriffsrechten ist selbstverständlich möglich. Um diese Methoden in Module einzubinden, muss die Datei "/functions/cp.ftp.php" per "include()" oder "include_once()" eingebunden werden.

6.8.1. Beispiel 1:

Mit diesem Beispiel soll der Funktionsumfang der Klasse demonstriert werden. Hier wird ein FTP-Server geöffnet und anschließend wird der Inhalt eines Unterordners als Array ausgegeben.

```
<?
// ...
// Contentpapst-Framework
// ...

include("functions/cp.ftp.php"); // Einbindung der Klasse

$ftp = new contentpapst_ftp();

$ftp->open("ftp.microsoft.com");

$ftp->change_dir("Products");
print_r($ftp->show());
$ftp->close();
?>
```

6.8.2. change_dir()

Mittels der Methode "change_dir()" wird ein Verzeichniswechsel auf einem FTP-Server initiiert. Dazu benötigt die Methode die Angabe des gewünschten Ordners zu dem gewechselt werden soll. Rückgabewerte sind je nach Ergebnis "1" (true) oder "0" (false):

```
$status = $ftp->change_dir("filme/komoedien"); // wechselt in das Verzeichnis "filme/komoedien"
```

6.8.3. chmod()

Wichtig für FTP-Server, die von mehreren Personen genutzt werden, ist die Sicherung von Dateien und Ordnern mittels CHMOD. Über den CHMOD kann detailliert beschrieben werden, welche Gruppen welche Rechte für eine Datei / einen Ordner haben. Für das Setzen des CHMOD steht die Methode "chmod()" zur Verfügung. Diese erhält als Parameter den Pfad zur Datei und den neuen CHMOD (3-stellig). Rückgabewerte sind je nach Ergebnis "1" (true) oder "0" (false):

```
$ftp->chmod("grafiken_2004.cdr", "777"); // gibt die Datei für alle Gruppen frei
```

6.8.4. close()

Über die Methode "close()" wird lediglich die über "open()" aufgebaute Verbindung zu einem FTP-Server wieder unterbrochen:

```
$ftp->close();
```

6.8.5. create_dir()

Die Methode "create_dir()" dient - wie könnte es anders sein - der Erstellung eines neuen Verzeichnisses. Hier wird ausgehend vom aktuellen Verzeichnis ein neues Verzeichnis mit dem als einzigen Parameter übergebenen Namen erstellt. Wenn bspw. ein Verzeichnis unterhalb des Startverzeichnisses erstellt werden soll, dann muss vorher mittels "change_dir ()" dorthin gewechselt werden. Standardmäßig wird versucht, dem neuen Verzeichnis den CHMOD 777 zu geben. Rückgabewerte sind, je nachdem ob das Verzeichnis erstellt werden konnte, "1" (true) oder "0" (false):

```
$ftp->create_dir("filme/komoedien"); // erstellt das Verzeichnis "filme/komoedien"
```

6.8.6. delete_dir()

Mit der Methode "delete_dir()" lassen sich schnell und einfach komplette Ordner mit Unterordnern und Dateien rekursiv löschen. Es muss dazu lediglich die Methode mit dem Namen des Verzeichnisses aufgerufen werden, in dem gestartet werden soll. Dieses Verzeichnis wird zum Schluss ebenfalls gelöscht.

```
$ftp->delete_dir("backup/2003/"); // löscht das komplette Verzeichnis "backup/2003"
```

6.8.7. delete_file()

Im Gegensatz zu "delete_dir()" wird mit der Methode "delete_file()" nur eine einzelne Datei gelöscht. Rückgabewerte sind je nach Ergebnis (Berechtigung vorhanden ja/nein) "1" (true) oder "0" (false):

```
$ftp->delete_file("backup/2003/vorlagen.zip"); // löscht die Datei "backup/2003/vorlagen.zip" vom Server
```

6.8.8. download_file()

Ein Download einer Datei vom FTP-Server kann über die Methode "download_file()" durchgeführt werden. Notwendig sind hier als Parameter (in dieser Reihenfolge) der Name für die lokale Datei, der Name der Datei auf dem Server und der Pfad zur Datei ausgehend vom aktuellen Verzeichnis. Mögliche Rückgabewerte sind "1" (Download erfolgreich), "-1" (es konnte nicht in den Ordner gewechselt werden), "-2" (Datei kann nicht heruntergeladen werden) und "-3" (Datei kann nicht geöffnet werden).

```
$ftp->download_file("lokal/download.gif", "screenshot_kunde.gif", "kunden03");
// lädt die Datei "screenshot_kunde.gif" als "lokal/download.gif" herunter
```

6.8.9. file_exists()

Ob eine Datei auf dem FTP-Server existiert, evaluieren Sie über die Methode "file_exists()". Diese benötigt als Parameter den Dateinamen und den Pfad zum Verzeichnis ausgehend vom aktuellen Verzeichnis. Rückgabewerte sind, je nachdem ob die Datei vorhanden ist, "1" (true) oder "0" (false):

```
$status = $ftp->file_exists("ibiza2003.jpg", "urlaubsfotos"); // überprüft, ob die Datei vorhanden ist
```

6.8.10. is_connected()

Mittels "is_connected()" kann überprüft werden, ob eine vorher aufgebaute Verbindung zu einem FTP-Server noch offen ist. Rückgabewerte sind je nach Ergebnis "1" (true) oder "0" (false):

```
$status = $ftp->is_connected(); // überprüft die aktuelle Verbindung
```

6.8.11. open()

Durch die Methode "open()" wird eine Verbindung zu einem FTP-Server hergestellt. Hierzu ist als Parameter vor allem der Host des FTP-Servers erforderlich. Da einige FTP-Server eine Authentifizierung erfordern oder unter einem anderen Port laufen, können als weitere Parameter optional Benutzername, Passwort und Port übergeben werden (in dieser Reihenfolge). Werden diese Parameter nicht übergeben, dann verwendet "open()" Standardwerte (Port 21 und Benutzername "anonymous").

Sobald die Methode alle notwendigen Werte erhalten hat, wird versucht eine Verbindung aufzubauen. Rückgabewerte sind je nach Ergebnis "1" (true) oder "0" (false):

```
$ftp->open("ftp.microsoft.com"); // stellt die Verbindung zu "ftp.microsoft.com" her
```

6.8.12. rename()

Die Methode "rename()" ermöglicht die Umbenennung von Dateien und Ordnern. Hierfür muss man sich bereits im richtigen Verzeichnis befinden. Der Methode werden dann als Parameter zuerst der alte und dann der neue Name der Datei / des Verzeichnisses übergeben. Rückgabewerte sind je nach Ergebnis "1" (true) oder "0" (false):

```
$ftp->rename("ibiza2003.jpg", "newzealand2003.jpg"); // gibt Datei wird in "newzealand2003.jpg" umbenannt
```

6.8.13. show()

Um die Methode "show()" zu nutzen, muss vorher bereits eine Verbindung mittels "open()" hergestellt wurden sein. Dann kann "show()" ein Pfad übergeben werden. Zurückgeliefert wird im Erfolgsfall ein Array mit allen Dateien und Ordnern in dieser Ebene (jeweils pro Eintrag - sofern vorhanden - Name, Größe, letzte Änderung und Typ). Wenn in den Ordner nicht gewechselt werden kann (z.B. wenn die Rechte dafür nicht ausreichend sind), wird von der Methode "0" (false) zurückgegeben. Wenn der Ordner hingegen leer ist, ist der Rückgabewert -1.

```
$status = $ftp->show("fussball/bilder/"); // liefert Array zu "fussball/bilder/"
```

6.8.14. upload_file()

Die Methode "upload_file()" geht genau den entgegengesetzten Weg zu "download_file()". Mit ihr ist es möglich eine Datei auf den Server zu verschieben. Die Parameter und Rückgabewerte bleiben die gleichen wie bei "download_file()".

```
$ftp->upload_file("lokal/download.gif", "screenshot_kunde.gif", "kunden03");
// lädt die Datei "lokal/download.gif" als "screenshot_kunde.gif" hoch
```

6.9. Funktionsbibliothek

In diesem Abschnitt erhalten Sie einen Überblick über die im System enthaltenen Funktionen, Klassen und Methoden. Aufgeführt werden (sofern verfügbar) neben den Namen der pro Datei verfügbaren Funktionen zusätzliche Beschreibungstexte und die jeweils zum Aufruf der Funktionen erforderlichen Parameter. Abhängig von der genutzten Lizenz müssen nicht alle der hier gelisteten Dateien und Funktionen vorhanden sein.

6.9.1. Datei "/config.dir.php"

Funktionen in der Datei "/config.dir.php" (Name der Funktion, Beschreibung, Parameter):

config_error()	Liefert eine Fehlermeldung aus, wenn die Datenbank und/oder das Vorlagenset nicht erreichbar sind.	$type

6.9.2. Datei "/functions.php"

Funktionen in der Datei "/functions.php" (Name der Funktion, Beschreibung, Parameter):

blocked()	Für den Zugriffsschutz werden veraltete Einträge entfernt.	
change_language()	Zeigt die Flaggen für den Wechsel zwischen den Sprachen an.	$mode = ""
cp_array_average()	Ermittelt den Durchschnitt der Zahlenwerte in einem Array.	$array, $start = "", $stop = ""
cp_array_structure()	Zeigt ein Array in einer Baumstruktur an.	$array, $level = "0"
cp_array_sum()	Summiert die Zahlenwerte in einem Array.	$array, $start = "", $stop = ""
cp_compress()	Führt die Komprimierung und optimierte Auslieferung der Seiten durch.	$level = "5", $debug = "0", $contents = ""
cp_compress_gzip()	Überprüft die für die GZIP-Komprimierung erforderlichen Einstellungen.	
cp_container_cache()	Erzeugt einen temporären Dateinamen für einen Container.	$form
cp_currency_info()	Liefert ein Array der Währungen nach ISO 4217 zurück.	$string = ""
cp_delete_cache()	Löscht eine temporäre Datei aus dem Cache.	$path, $filename
cp_download_file()	Lädt eine via $filename übermittelte Datei herunter.	$filename, $file_length = "", $mime_type = "application/octet-stream", $cache = "1"
cp_error_log()	Fängt Fehlermeldungen des Systems ab und setzt einen neuen Logeintrag.	$errno, $errmsg, $filename, $linenum
cp_escape_array()	Überprüft die Angaben in einem Array vor der Übergabe in die Datenbank.	$array
cp_file_type_image()	Gibt das zugehörige Bild für einen Dateityp zurück.	$type
cp_file_upload()	Lädt eine einzelne Datei auf dem Server in einen angegebenen Ordner.	$path, $file
cp_form_tunnel()	Erstellt einen Form-Tunnel für die Übergabe von Variablen.	$type, $params
cp_form_tunnel_breaks ()	Reinigt die Variablen für die Übermittelung mittels eines Form-Tunnels.	$string
cp_form_tunnel_clean()	Passt die über einen Form-Tunnel übermittelten Variablen an.	$string
cp_get_cache()	Liest den Status einer temporären Datei im Cache.	$path, $filename, $time_max = "3600", $delete_oldfile = "1"
cp_get_country()	Liefert eine Angabe zu einem einzelnen Land zurück (Domainkürzel, Titel).	$number, $field = "title"
cp_get_custom_fields()	Liefert die Namen der benutzerspezischen Felder einer Datenbanktabelle als Array zurück.	$table, $prefix = "custom_"
cp_header_cget()	Setzt den ETag-Header für das Datum der letzten Änderung der Seite.	$timestamp
cp_include()	Bindet eine Text-, HTML- oder PHP-Datei ein.	$url, $type = ""
cp_is_editable()	Überprüft, ob Dateityp in der Dateiverwaltung bearbeitet werden kann.	$file_type
cp_is_inside()	Überprüft, ob sich ein Aufruf innerhalb des Systems befindet.	$string
cp_key_exists()	Überprüft, ob ein Schlüssel in einem Array vorhanden ist.	$array, $key
cp_lang_info()	Liefert ein Array der Sprachen nach ISO 639-1 zurück.	$string = ""
cp_magic_metatags()	Erstellt automatisch Schlüsselwörter für die Metatags aus einem Text.	$content, $max_metatags = ""
cp_main_header()	Erzeugt die obere Breadcrumb-Navigation für ein Modul.	$basics, $array
cp_meta_tags()	Sorgt für die richtige Verwendung der im System eingegebenen Metatag-Angaben.	$content
cp_mime_info()	Liefert Informationen über den MIME-Typ eines Dateiformates zurück.	$string = ""
cp_module_exists()	Überprüft, ob ein Modul im System vorhanden ist.	$name
cp_php_version()	Ermittelt die auf dem Server zur Verfügung stehende PHP-Version.	
cp_print_r()	Funktions-Wrapper für die Standard-Funktion print_r().	$array
cp_random_string()	Erzeugt einen zufallsgenerierten String beliebiger Länge.	$length
cp_read_cache()	Lädt eine temporäre Datei aus dem Cache.	$path, $filename, $return = "0"
cp_return_request()	Wechselt nach dem Aufruf einer Funktion auf eine bestimmte URL, sofern der Parameter "cleanajax" in der URL fehlt.	$url, $http_status = ""
cp_set_cache()	Schreibt eine temporäre Datei in den Cache.	$input, $path, $filename
cp_set_custom_fields()	Hinterlegt den Inhalt der benutzerspezischen Felder in der Datenbanktabelle eines Modules.	$table, $field = "number", $array
cp_set_log()	Fügt einen neuen Eintrag zum Log des Systems hinzu (z.B. Fehlermeldung).	$name, $message = "", $module = "", $number = ""
cp_show_custom_fields ()	Gibt den Inhalt der benutzerspezischen Felder in den Formularen aus. Aufruf: "table" (Name der Datenbanktabelle ohne Präfix, "array" (Array mit Angaben eines Datensatzes), "prepared" (Array mit vorgefertigen Formularfeldern für einzelne	$table, $array = array(), $prepared = array(), $clean = "0"

	Angaben, die z.B. eine spezielle Formatierung benötigen).	
cp_siteinfo()	Liefert Informationen über einen Mandanten anhand des internen Namens zurück.	$name, $field
cp_sql_split()	Zerteilt eine angelieferte Datei in die einzelnen SQL-Befehle.	$string
cp_string_format()	Konvertiert Sonderzeichen in das Unicode-Format.	$text, $specialchars = "1"
cp_urlencode_array()	URL-kodiert die Angaben in einem Array vor der Übergabe per URL.	$array
cp_validate_color()	Validiert eine Farbangabe im Standardformat (z.B. "#FFFFFF").	$query
cp_validate_email()	Validiert eine Email-Adresse im Standardformat.	$string
cp_validate_number()	Überprüft, ob der übergebene Wert eine Zahl ist.	$number
cp_validate_string()	Überprüft, ob der übergebene Wert den Kriterien entspricht.	$string, $min = "", $max = ""
cp_word_shorten()	Verkürzt die maximale Länge von Worten in einem Text.	$string, $length = "50"
database_in()	Überprüft einen Text, bevor dieser in die Datenbank eingefügt wird.	$text, $options = array()
form_file()	Zeigt das Formularelement für den Dateiupload an. Parameter: "action", "method", "name"	$action, $method = "post", $name = "form_standard"
form_hidden()	Zeigt ein verstecktes Formularelement an. Parameter: "name", "value", "id"	$name, $value, $id = ""
form_moveonsite()	Zeigt das Formularelement für einen Button zum Wechsel auf eine neue Seite an. Parameter "target", "text", "relocate", "onclick"	$target, $value, $relocate = "1", $onclick = "" $class = "inputsubmit"
form_send()	Zeigt das Formularelement zum Abschicken des Formulars an. Parameter: "onclick", "button", "value"	$onclick = "", $button = "", $value = ""
form_start()	Wrapper-Funktionen für die Generierung von Formularelementen.	$action, $method = "post", $name = "form_standard"
form_start_wysiwyg()	Wrapper-Funktionen für die Generierung von Formularelementen mit einem WYSIWYG-Editor.	$action, $method = "post", $name = "form_standard", $enctype = ""
form_stop()	Schließt ein über form_start() bzw. form_start_wysiwyg() geöffnetes Formular ab.	
form_text()	Zeigt das Formularelement für die Eingabe eines einzeiligen Textes an. Parameter: "name", "class", "value", "length", "id", "disabled"	$name, $class, $value, $length, $id = "", $disabled = ""
form_textarea()	Zeigt das Formularelement für die Eingabe eines mehrzeiligen Textes an. Parameter "name", "class", "rows", "value", "id", "disabled"	$name, $class, $rows, $value, $id = "", $disabled = ""
form_url()	Passt einzelne Zeichen in einer URL für XHTML an.	$string
form_wysiwyg()	Zeigt das Formularelement für den WYSIWYG-Editor an. Parameter: "options", "name", "force_textarea"	$options, $name, $force_textarea = "0"
form_wysiwyg_clean()	Säubert die per WYSIWYG-Editor generierten Inhalte.	$string
get_dir()	Ermittelt die Dateien in einem einzelnen Ordner.	$sPath, $var, $return = ""
get_random()	Ermittelt zufällig einen Eintrag aus der Datenbanktabelle "*_random".	$type
image_nav()	Generiert eine Navigation auf der Basis einer Grafik.	$quality = "75"
js_check()	Erzeugt eine Javascript-Funktion zur Formular-Validierung.	$form, $fields, $alert, $function_name = "check_form", $return = "", $return_string = "", $insert_after = "", $insert_before = "", $insert_last = ""
linkshyper()	Nimmt Formatierungen an einem Text vor (Links und Emails markieren etc.).	$text, $mode
mod_rewrite_name()	Erzeugt einen für mod_rewrite geeigneten Dateinamen.	$number, $title, $module = "article", $onlyname = ""
mod_rewrite_string()	Erzeugt einen für mod_rewrite geeigneten Dateinamen.	$string
module_access_restricted ()	Gibt eine Fehlermeldung bei unberechtigtem Zugriff aus.	
module_language()	Lädt die Sprachdatei zu einem gewünschten Modul.	$module = ""
revert_htmlspecialchars ()	Kehrt die Wirkungsweise der PHP-Funktion htmlspecialchars() um.	$string
sendsql()	Übermittelt SQL-Befehle an die Datenbank.	$type, $filename
simplexml2array()	Wandelt ein SimpleXML-Objekt in ein Array um.	$xml
simplexml2load()	Lädt eine XML-Datei und liefert diese als Array zurück.	$file, $cache = "0", $is_string = "0"
tablebreak()	Fügt in eine Tabelle einen zusätzlichen Absatz ein, um die Übersichtlichkeit zu erhöhen.	$comment = "", $padding = "1"
tablefooter()	Erstellt das Ende einer Tabelle auf der Basis einer Vorlage.	$title = ""
tablenavigation()	Erstellt die Navigation beim Bearbeiten von Einträgen mit den Vor-/Zurück-Links und der aktuellen Position in den Einträgen.	$table, $number, $url, $query = "", $field = "number"
tablepaging()	Erstellt die Blätternavigation unterhalb einer Tabelle, "[ID]" in der URL wird durch die jeweilige Seitennummer ersetzt.	$url, $total, $current = 1, $show_max = 8, $return = "0"
tabletop()	Erstellt den Beginn einer Tabelle auf der Basis einer Vorlage.	$title = "", $cleanup = "1", $type = "1", $edit_control = "", $refresh_control = "", $refresh_url = ""
tools_getmicrotime()	Timer-Funktionen für die Überprüfung der zur Generierung der Seite benötigten Zeit.	
tools_timerStart()	Startet den Timer für die Überprüfung der zur Generierung der Seite benötigten Zeit.	
tools_timerStop()	Stopt den Timer für die Überprüfung der zur Generierung der Seite benötigten Zeit.	$decimals = "4"
validate_email()	Überprüfung des Formats einer Email-Adresse.	$address
visitorsonline()	Ermittelt die Anzahl der momentanen Besucher auf der Seite.	$seconds = 1800

6.9.3. Datei "/install.php"

Funktionen in der Datei "/install.php" (Name der Funktion, Beschreibung, Parameter):

install_check()	- Keine Beschreibung verfügbar. -	$type
install_connect()	- Keine Beschreibung verfügbar. -	$form
install_db()	- Keine Beschreibung verfügbar. -	$error = "", $data = ""
install_error()	- Keine Beschreibung verfügbar. -	$error
install_get_db()	- Keine Beschreibung verfügbar. -	$path
install_license()	- Keine Beschreibung verfügbar. -	
install_main()	- Keine Beschreibung verfügbar. -	$error = "", $data = ""
install_setup()	- Keine Beschreibung verfügbar. -	$contents, $form
install_step_1()	- Keine Beschreibung verfügbar. -	$form
install_step_2()	- Keine Beschreibung verfügbar. -	$form
install_step_3()	- Keine Beschreibung verfügbar. -	
install_step_4()	- Keine Beschreibung verfügbar. -	$form
install_step_5()	- Keine Beschreibung verfügbar. -	

6.9.4. Datei "/update.php"

Funktionen in der Datei "/update.php" (Name der Funktion, Beschreibung, Parameter):

changes()	- Keine Beschreibung verfügbar. -	$old_file, $new_file
check_errors()	- Keine Beschreibung verfügbar. -	$file, $file_title, $array
compare()	- Keine Beschreibung verfügbar. -	$old_file, $new_file, $linepadding = null
db_close()	- Keine Beschreibung verfügbar. -	$query
db_connect()	- Keine Beschreibung verfügbar. -	$database, $server, $user = "", $password = "", $port = "", $socket = ""
db_escape_string()	- Keine Beschreibung verfügbar. -	$string
db_fetch_array()	- Keine Beschreibung verfügbar. -	$query, $type = "both"
db_fetch_object()	- Keine Beschreibung verfügbar. -	$query
db_list_dbs()	- Keine Beschreibung verfügbar. -	$query
db_list_tables()	- Keine Beschreibung verfügbar. -	$query, $connection
db_num_rows()	- Keine Beschreibung verfügbar. -	$query
db_query()	- Keine Beschreibung verfügbar. -	$query
db_select_db()	- Keine Beschreibung verfügbar. -	$query, $db_link = ""
db_split()	Zerteilt eine angelieferte Datei in die einzelnen SQL-Befehle.	$string
flush()	- Keine Beschreibung verfügbar. -	
form_button()	- Keine Beschreibung verfügbar. -	$name, $title, $query = "", $update_options = array()
form_checkbox()	- Keine Beschreibung verfügbar. -	$name, $id, $value, $checked, $string, $type = ""
form_start()	- Keine Beschreibung verfügbar. -	$name
form_stop()	- Keine Beschreibung verfügbar. -	$name, $title, $count = ""
format()	- Keine Beschreibung verfügbar. -	$code
format_line()	- Keine Beschreibung verfügbar. -	$string
get_cpforms()	- Keine Beschreibung verfügbar. -	
get_database_fields()	- Keine Beschreibung verfügbar. -	$table
get_database_tables()	- Keine Beschreibung verfügbar. -	
get_directories()	- Keine Beschreibung verfügbar. -	$directory = ".", $array = array()
get_files()	- Keine Beschreibung verfügbar. -	$directory = "."
get_sites()	- Keine Beschreibung verfügbar. -	
layout_footer()	- Keine Beschreibung verfügbar. -	
layout_header()	- Keine Beschreibung verfügbar. -	$options = array()
show_config()	- Keine Beschreibung verfügbar. -	$array
show_message()	- Keine Beschreibung verfügbar. -	$string, $type = "", $header = ""
showline()	- Keine Beschreibung verfügbar. -	$line
simplexml2array()	Wandelt ein SimpleXML-Objekt in ein Array um.	$xml
simplexml2load()	Lädt eine XML-Datei und liefert diese als Array zurück.	$file
wrap()	- Keine Beschreibung verfügbar. -	$old_file, $new_file

6.9.5. Datei "/database/db.mysql.php"

Funktionen in der Datei "/database/db.mysql.php" (Name der Funktion, Beschreibung, Parameter):

db_close()	- Keine Beschreibung verfügbar. -	$query
db_connect()	- Keine Beschreibung verfügbar. -	$database, $server, $user = "", $password = "", $port = "", $socket = ""
db_create_db()	- Keine Beschreibung verfügbar. -	$query
db_delete()	- Keine Beschreibung verfügbar. -	$query
db_drop_db()	- Keine Beschreibung verfügbar. -	$query
db_escape_string()	- Keine Beschreibung verfügbar. -	$string
db_fetch_array()	- Keine Beschreibung verfügbar. -	$query, $type = "both"

db_fetch_object()	- Keine Beschreibung verfügbar. -	$query
db_fetch_row()	- Keine Beschreibung verfügbar. -	$query
db_field_flags()	- Keine Beschreibung verfügbar. -	$query, $i
db_field_len()	- Keine Beschreibung verfügbar. -	$query, $i
db_field_name()	- Keine Beschreibung verfügbar. -	$query, $i
db_field_type()	- Keine Beschreibung verfügbar. -	$query, $i
db_free_result()	- Keine Beschreibung verfügbar. -	$query
db_insert()	- Keine Beschreibung verfügbar. -	$query
db_last_id()	- Keine Beschreibung verfügbar. -	$result = ""
db_list_dbs()	- Keine Beschreibung verfügbar. -	$query
db_list_fields()	- Keine Beschreibung verfügbar. -	$query, $table, $connection, $action = ""
db_list_tables()	- Keine Beschreibung verfügbar. -	$query, $connection
db_num_fields()	- Keine Beschreibung verfügbar. -	$query
db_num_rows()	- Keine Beschreibung verfügbar. -	$query
db_optimize()	- Keine Beschreibung verfügbar. -	$query
db_select()	- Keine Beschreibung verfügbar. -	$query
db_select_db()	- Keine Beschreibung verfügbar. -	$query, $db_link = ""
db_update()	- Keine Beschreibung verfügbar. -	$query
db_version()	- Keine Beschreibung verfügbar. -	

6.9.6. Datei "/functions/class.database.php"

Funktionen in der Datei "/functions/class.database.php" (Name der Funktion, Beschreibung, Parameter):

delete_entry()	Entfernt einen bestehenden Eintrag aus der Datenbank. Aufruf: $GLOBALS["cpclass"]->database->delete_entry("article", "123"); // Entfernt den Artikel mit der ID 123 aus der Datenbank.	$table, $number
insert_entry()	Fügt einen neuen Eintrag in die Datenbank ein. Aufruf: $GLOBALS["cpclass"]->database->insert_entry ("article", array("field"=>"value", "field_2"=>"...")); // Liefert die ID des neuen Eintrages zurück.	$table, $array
select_entry()	Lädt einen kompletten Eintrag oder eine einzelne Information eines Eintrages aus der Datenbank. Aufruf: $variable = $GLOBALS["cpclass"]->database->select_entry("article", "123", "title"); // Liefert den Titel des Artikels mit der ID 123 zurück.	$table, $number, $field = "", $query = ""
update_entry()	Aktualisiert einen bestehenden Eintrag in der Datenbank. Aufruf: $GLOBALS["cpclass"]->database->update_entry("article", "123", array("field"=>"value", "field_2"=>"...")); // Aktualisiert den Artikel mit der ID 123.	$table, $number, $array, $query = ""

6.9.7. Datei "/functions/class.forms.php"

Funktionen in der Datei "/functions/class.forms.php" (Name der Funktion, Beschreibung, Parameter):

back()	Zeigt das Formularelement für den Zurück-Button an (Wechsel per Javascript). Parameter im Array: "class", "return", "steps", "value" Aufruf: $GLOBALS["cpclass"]->forms->back("...");	$array = array()
captcha()	Zeigt das Formularelement für den grafischen Sicherheitscode an (Captcha). Parameter im Array: "id", "name", "return" Aufruf: $GLOBALS["cpclass"]->forms->captcha(array("..."=>"..."));	$array
captcha_validate()	Überprüft den vom Benutzer eingegebenen Wert für den grafischen Sicherheitscode (Captcha) und bricht gegebenenfalls die Verarbeitung ab. Parameter im Array: "name", "value", "url" Aufruf: $GLOBALS["cpclass"]->forms->captcha_validate (array("..."=>"..."));	$array
checkbox()	Zeigt das Formularelement für die Kontrollkästchen an. Parameter im Array: "checked", "disabled", "id", "name", "onclick", "return", "value" Aufruf: $GLOBALS["cpclass"]->forms->checkbox(array("..."=>"..."));	$array
move()	Zeigt das Formularelement für einen Button zum Wechsel auf eine neue Seite an. Parameter im Array "action", "class", "onclick", "relocate", "return", "value" Aufruf: $GLOBALS["cpclass"]->forms->move(array("..."=>"..."));	$array
password()	Zeigt das Formularelement für die Passwort-Eingabe an. Parameter im Array: "class", "id", "length", "name", "return", "value" Aufruf: $GLOBALS["cpclass"]->forms->password(array("..."=>"..."));	$array
radio()	Zeigt das Formularelement für die Optionsfelder an. Parameter im Array: "checked", "id", "name", "return", "value" Aufruf: $GLOBALS["cpclass"]->forms->radio(array("..."=>"..."));	$array
reset()	Zeigt das Formularelement zum Zurücksetzen des Formularinhalts an. Parameter im Array: "class", "return", "value" Aufruf: $GLOBALS["cpclass"]->forms->reset(array("..."=>"..."));	$array = array()

6.9.8. Datei "/functions/class.template.php"

Funktionen in der Datei "/functions/class.template.php" (Name der Funktion, Beschreibung, Parameter):

assign()	- Keine Beschreibung verfügbar. -	$string
change()	- Keine Beschreibung verfügbar. -	$type, $variable, $content
clean()	- Keine Beschreibung verfügbar. -	
dump()	- Keine Beschreibung verfügbar. -	$file, $string
format()	- Keine Beschreibung verfügbar. -	$string
format_client()	- Keine Beschreibung verfügbar. -	$params, $content
format_container()	- Keine Beschreibung verfügbar. -	
format_date()	- Keine Beschreibung verfügbar. -	$format, $time= ""
format_language()	- Keine Beschreibung verfügbar. -	$module, $value

format_texteffect()	- Keine Beschreibung verfügbar. -	$string
open()	- Keine Beschreibung verfügbar. -	$template, $mode = "complete", $is_file = "1", $parse_container = "1", $template_options = ""
output()	- Keine Beschreibung verfügbar. -	$return = "", $clean_ignore = "1", $safe_tags = ""
output_tags()	- Keine Beschreibung verfügbar. -	$return = "1"
preplace()	- Keine Beschreibung verfügbar. -	$var, $string
process()	- Keine Beschreibung verfügbar. -	$content, $mode = "complete", $parse_container = "1", $template_options = ""
process_current()	- Keine Beschreibung verfügbar. -	
replace()	- Keine Beschreibung verfügbar. -	$var, $string
rewrite()	- Keine Beschreibung verfügbar. -	$string
rights()	- Keine Beschreibung verfügbar. -	$string, $content
template()	- Keine Beschreibung verfügbar. -	$string, $content

6.9.9. Datei "/functions/cp.css.php"

Funktionen in der Datei "/functions/cp.css.php" (Name der Funktion, Beschreibung, Parameter):

cp_css_input()	- Keine Beschreibung verfügbar. -	$file
cp_css_output()	- Keine Beschreibung verfügbar. -	$file, $tag_array

6.9.10. Datei "/functions/cp.date.php"

Funktionen in der Datei "/functions/cp.date.php" (Name der Funktion, Beschreibung, Parameter):

check_date()	- Keine Beschreibung verfügbar. -	$date
days_per_month()	- Keine Beschreibung verfügbar. -	$month = "", $year = ""
first_monday()	- Keine Beschreibung verfügbar. -	$year = ""
is_leap_year()	- Keine Beschreibung verfügbar. -	$year = ""
is_valid_date()	- Keine Beschreibung verfügbar. -	$day, $month, $year
mysql_timestamp()	- Keine Beschreibung verfügbar. -	
week_monday()	- Keine Beschreibung verfügbar. -	$week, $year = ""

6.9.11. Datei "/functions/cp.files.php"

Funktionen in der Datei "/functions/cp.files.php" (Name der Funktion, Beschreibung, Parameter):

delete_dirs()	- Keine Beschreibung verfügbar. -	$array
delete_files()	- Keine Beschreibung verfügbar. -	$array
directories()	- Keine Beschreibung verfügbar. -	$dir
extension()	- Keine Beschreibung verfügbar. -	$file
files()	- Keine Beschreibung verfügbar. -	$array
get_files()	- Keine Beschreibung verfügbar. -	$dir
list_dirs()	- Keine Beschreibung verfügbar. -	$dir, $subdirs = "1"
list_files()	- Keine Beschreibung verfügbar. -	$dir, $subdirs = "0"

6.9.12. Datei "/functions/cp.ftp.php"

Funktionen in der Datei "/functions/cp.ftp.php" (Name der Funktion, Beschreibung, Parameter):

change_dir()	- Keine Beschreibung verfügbar. -	$path
chmod()	- Keine Beschreibung verfügbar. -	$file, $rights
close()	- Keine Beschreibung verfügbar. -	
create_dir()	- Keine Beschreibung verfügbar. -	$foldername
delete_dir()	- Keine Beschreibung verfügbar. -	$dir
delete_file()	- Keine Beschreibung verfügbar. -	$file
download_file()	- Keine Beschreibung verfügbar. -	$file, $filename, $path
file_exists()	- Keine Beschreibung verfügbar. -	$file, $path
is_connected()	- Keine Beschreibung verfügbar. -	
open()	- Keine Beschreibung verfügbar. -	$host, $username = "anonymous", $password = "", $port = "21"
rename()	- Keine Beschreibung verfügbar. -	$old, $new
show()	- Keine Beschreibung verfügbar. -	$path = ""
upload_file()	- Keine Beschreibung verfügbar. -	$file, $filename, $path

6.9.13. Datei "/functions/cp.graph.php"

Funktionen in der Datei "/functions/cp.graph.php" (Name der Funktion, Beschreibung, Parameter):

add_point()	- Keine Beschreibung	$point

	verfügbar. -	
add_spline()	- Keine Beschreibung verfügbar. -	$title, $type, $color, $array
calculate_allborders()	- Keine Beschreibung verfügbar. -	
calculate_borders()	- Keine Beschreibung verfügbar. -	
calculate_margin()	- Keine Beschreibung verfügbar. -	
calculate_tics()	- Keine Beschreibung verfügbar. -	
calculate_values()	- Keine Beschreibung verfügbar. -	
contentpapst_axis()	- Keine Beschreibung verfügbar. -	$min, $max
contentpapst_bar_diagram ()	- Keine Beschreibung verfügbar. -	$width, $height
contentpapst_color()	- Keine Beschreibung verfügbar. -	$red, $green, $blue, $alpha = 0
contentpapst_diagram()	- Keine Beschreibung verfügbar. -	$width, $height
contentpapst_graph()	- Keine Beschreibung verfügbar. -	$width, $height, $border_horizontal, $border_vertical, $border_color, $bgcolor
contentpapst_horizontal_diagram ()	- Keine Beschreibung verfügbar. -	$width = 600, $height = 300
contentpapst_line_diagram ()	- Keine Beschreibung verfügbar. -	$width = 600, $height = 300
contentpapst_pie_diagram ()	- Keine Beschreibung verfügbar. -	$width = 600, $height = 300, $show_logo = 1, $show_title = 1, $show_label = 1, $show_background = 1
contentpapst_point()	- Keine Beschreibung verfügbar. -	$x, $y
contentpapst_primitive()	- Keine Beschreibung verfügbar. -	$img
contentpapst_text()	- Keine Beschreibung verfügbar. -	
contentpapst_vertical_diagram ()	- Keine Beschreibung verfügbar. -	$width = 600, $height = 300
create_image()	- Keine Beschreibung verfügbar. -	
create_splines()	- Keine Beschreibung verfügbar. -	
create_table()	- Keine Beschreibung verfügbar. -	
create_wire()	- Keine Beschreibung verfügbar. -	
draw_disc()	- Keine Beschreibung verfügbar. -	$cy, $colorArray, $mode
draw_values()	- Keine Beschreibung verfügbar. -	
get_border_bottom()	- Keine Beschreibung verfügbar. -	
get_border_top()	- Keine Beschreibung verfügbar. -	
get_color()	- Keine Beschreibung verfügbar. -	$img
get_tics()	- Keine Beschreibung verfügbar. -	
get_x()	- Keine Beschreibung verfügbar. -	
get_y()	- Keine Beschreibung verfügbar. -	
line()	- Keine Beschreibung verfügbar. -	$x1, $y1, $x2, $y2, $color, $width = 1
outlined_box()	- Keine Beschreibung verfügbar. -	$x1, $y1, $x2, $y2
render_axis()	- Keine Beschreibung verfügbar. -	
render_bar()	- Keine Beschreibung verfügbar. -	
render_centered()	- Keine Beschreibung verfügbar. -	$img, $py, $color, $text, $file_name
render_diagonal()	- Keine Beschreibung verfügbar. -	$img, $px, $py, $color, $text
render_label()	- Keine Beschreibung verfügbar. -	

render_line()	- Keine Beschreibung verfügbar. -	
render_logo()	- Keine Beschreibung verfügbar. -	
render_pie()	- Keine Beschreibung verfügbar. -	
render_text()	- Keine Beschreibung verfügbar. -	$img, $px, $py, $color, $text, $file_name, $align = 0
render_title()	- Keine Beschreibung verfügbar. -	
reset()	- Keine Beschreibung verfügbar. -	
set_border_bottom()	- Keine Beschreibung verfügbar. -	$lowerBound
set_border_top()	- Keine Beschreibung verfügbar. -	$upperBound
set_borders()	- Keine Beschreibung verfügbar. -	$sampleCount, $yMinValue, $yMaxValue
set_logo()	- Keine Beschreibung verfügbar. -	$file_name
set_margin()	- Keine Beschreibung verfügbar. -	$margin
set_margin_bottom()	- Keine Beschreibung verfügbar. -	$labelMarginBottom
set_margin_center()	- Keine Beschreibung verfügbar. -	$labelMarginCenter
set_margin_left()	- Keine Beschreibung verfügbar. -	$labelMarginLeft
set_margin_right()	- Keine Beschreibung verfügbar. -	$labelMarginRight
set_margin_top()	- Keine Beschreibung verfügbar. -	$labelMarginTop
set_pie_ratio()	- Keine Beschreibung verfügbar. -	$pieRatio
set_title()	- Keine Beschreibung verfügbar. -	$title
show()	- Keine Beschreibung verfügbar. -	$file_name = null
sort_pie()	- Keine Beschreibung verfügbar. -	$v1, $v2

6.9.14. Datei "/functions/cp.image.php"

Funktionen in der Datei "/functions/cp.image.php" (Name der Funktion, Beschreibung, Parameter):

add_watermark()	- Keine Beschreibung verfügbar. -	$watermark
brightness_contrast()	- Keine Beschreibung verfügbar. -	$brightness, $contrast
brightness_helper()	- Keine Beschreibung verfügbar. -	$rgb, $brightness, $contrast
colorize()	- Keine Beschreibung verfügbar. -	$color, $percent = "50"
colorpalette()	- Keine Beschreibung verfügbar. -	$image, $limit = "50"
create()	- Keine Beschreibung verfügbar. -	$width, $height, $type
gd_version()	- Keine Beschreibung verfügbar. -	
get_colorcode()	- Keine Beschreibung verfügbar. -	$code, $mode = "1"
get_quality()	- Keine Beschreibung verfügbar. -	
get_type()	- Keine Beschreibung verfügbar. -	
grayscale()	- Keine Beschreibung verfügbar. -	
invert()	- Keine Beschreibung verfügbar. -	
monochrome()	- Keine Beschreibung verfügbar. -	
open()	- Keine Beschreibung verfügbar. -	$filename
piechart()	- Keine Beschreibung verfügbar. -	$radius = "100", $labels, $width, $colors
resize()	- Keine Beschreibung verfügbar. -	$width,$height,$truecolor = "1"
rotate()	- Keine Beschreibung verfügbar. -	$degrees = "0"
save()	- Keine Beschreibung verfügbar. -	$filename = "", $block_header = "0"
set_quality()	- Keine Beschreibung verfügbar. -	$quality
set_type()	- Keine Beschreibung verfügbar. -	$type
show()	- Keine Beschreibung verfügbar. -	

6.9.15. Datei "/functions/cp.mail.php"

Funktionen in der Datei "/functions/cp.mail.php" (Name der Funktion, Beschreibung, Parameter):

add_attachment()	- Keine Beschreibung verfügbar. -	$file, $type = "", $encoding = "", $disposition = "", $description = "", $name = ""

add_bcc()	- Keine Beschreibung verfügbar. -	$email
add_cc()	- Keine Beschreibung verfügbar. -	$email
clean_email()	- Keine Beschreibung verfügbar. -	$string
contentpapst_mail()	- Keine Beschreibung verfügbar. -	
convert_text()	- Keine Beschreibung verfügbar. -	
convert_to_text()	- Keine Beschreibung verfügbar. -	$string
create_header()	- Keine Beschreibung verfügbar. -	
encode_quotedprintable ()	- Keine Beschreibung verfügbar. -	$string
get_attachment()	- Keine Beschreibung verfügbar. -	$file
get_bcc()	- Keine Beschreibung verfügbar. -	$formatted = ""
get_cc()	- Keine Beschreibung verfügbar. -	$formatted = ""
receive_receipt()	- Keine Beschreibung verfügbar. -	
send()	- Keine Beschreibung verfügbar. -	$mode = ""
send_smtp()	- Keine Beschreibung verfügbar. -	$host, $port = "25", $username = "", $password = "", $mail_data = array()
set_bcc()	- Keine Beschreibung verfügbar. -	$email
set_body()	- Keine Beschreibung verfügbar. -	$mail_body
set_cc()	- Keine Beschreibung verfügbar. -	$email
set_charset()	- Keine Beschreibung verfügbar. -	$charset
set_from()	- Keine Beschreibung verfügbar. -	$mail_from, $mail_name = ""
set_ip()	- Keine Beschreibung verfügbar. -	$mail_ip = ""
set_mime_types()	- Keine Beschreibung verfügbar. -	$array
set_options()	- Keine Beschreibung verfügbar. -	$array
set_priority()	- Keine Beschreibung verfügbar. -	$mail_priority
set_reply_to()	- Keine Beschreibung verfügbar. -	$email
set_subject()	- Keine Beschreibung verfügbar. -	$mail_subject
set_to()	- Keine Beschreibung verfügbar. -	$mail_to, $mail_name = ""
set_type()	- Keine Beschreibung verfügbar. -	$mail_type
set_x_bounced()	- Keine Beschreibung verfügbar. -	$string
set_x_mailer()	- Keine Beschreibung verfügbar. -	$string
show_header()	- Keine Beschreibung verfügbar. -	
validate_email()	- Keine Beschreibung verfügbar. -	$email

6.9.16. Datei "/functions/cp.misc.php"

Funktionen in der Datei "/functions/cp.misc.php" (Name der Funktion, Beschreibung, Parameter):

cp_clean_indents()	- Keine Beschreibung verfügbar. -	$string
cp_clean_li()	- Keine Beschreibung verfügbar. -	$string
cp_clean_parameter()	- Keine Beschreibung verfügbar. -	$parameter, $value
cp_clean_tags()	- Keine Beschreibung verfügbar. -	$tag, $string
cp_clean_url()	- Keine Beschreibung verfügbar. -	$string
cp_clean_wysiwyg()	Reinigt die aus den WYSIWYG-Editoren kommenden Inhalte, bevor diese in der Datenbank gespeichert werden.	$string, $options = ""
cp_convert_filesize()	- Keine Beschreibung verfügbar. -	$filesize, $return, $decimals = "2"
cp_extract_exif()	- Keine Beschreibung verfügbar. -	$file_name
cp_extract_id3()	- Keine Beschreibung verfügbar. -	$file_name
cp_extract_iptc()	- Keine Beschreibung verfügbar. -	$file_name
cp_extract_pdf()	- Keine Beschreibung verfügbar. -	$file_name
cp_get_charset()	- Keine Beschreibung verfügbar. -	$language
cp_js_alert()	- Keine Beschreibung verfügbar. -	$string, $return
cp_request_uri()	- Keine Beschreibung verfügbar. -	

6.9.17. Datei "/functions/cp.navigation.php"

Funktionen in der Datei "/functions/cp.navigation.php" (Name der Funktion, Beschreibung, Parameter):

cp_navigation_children()	- Keine Beschreibung verfügbar. -	$number
cp_navigation_get_root ()	- Keine Beschreibung verfügbar. -	$number, $level = "0"
cp_navigation_is_branch ()	- Keine Beschreibung verfügbar. -	$number
cp_navigation_is_root()	- Keine Beschreibung verfügbar. -	$number
cp_navigation_list()	- Keine Beschreibung verfügbar. -	$number = ""
cp_navigation_parent()	- Keine Beschreibung verfügbar. -	$number
cp_navigation_title()	- Keine Beschreibung verfügbar. -	$number
cp_navigation_url()	- Keine Beschreibung verfügbar. -	$number

6.9.18. Datei "/functions/cp.pdf.php"

Funktionen in der Datei "/functions/cp.pdf.php" (Name der Funktion, Beschreibung, Parameter):

cp_pdf_prepare()	Bereitet Inhalte für die Ausgabe im PDF-Format vor.	$content

6.9.19. Datei "/functions/cp.rtf.php"

Funktionen in der Datei "/functions/cp.rtf.php" (Name der Funktion, Beschreibung, Parameter):

convert()	- Keine Beschreibung verfügbar. -	$mode = ""
create()	- Keine Beschreibung verfügbar. -	$mode = ""
format_colors()	- Keine Beschreibung verfügbar. -	
open()	- Keine Beschreibung verfügbar. -	$filename
save()	- Keine Beschreibung verfügbar. -	$filename
set_author()	- Keine Beschreibung verfügbar. -	$author
set_capitals()	- Keine Beschreibung verfügbar. -	
set_company()	- Keine Beschreibung verfügbar. -	$company
set_content()	- Keine Beschreibung verfügbar. -	$string
set_create_time()	- Keine Beschreibung verfügbar. -	$timestamp = ""
set_rev_time()	- Keine Beschreibung verfügbar. -	$timestamp = ""
set_title()	- Keine Beschreibung verfügbar. -	$title

6.9.20. Datei "/functions/cp.search.php"

Funktionen in der Datei "/functions/cp.search.php" (Name der Funktion, Beschreibung, Parameter):

cp_check_form()	- Keine Beschreibung verfügbar. -	$input, $search_min_length = "3", $search_max_length = "20"
cp_search_highlight()	- Keine Beschreibung verfügbar. -	$string, $search, $color
cp_search_result()	- Keine Beschreibung verfügbar. -	$query, $tables, $show_fields, $name_fields, $unique, $limit_length, $use_html, $template, $options = array()
cp_search_table()	- Keine Beschreibung verfügbar. -	$table, $fields, $show_fields, $name_fields, $query, $options = array()

6.9.21. Datei "/functions/cp.session.php"

Funktionen in der Datei "/functions/cp.session.php" (Name der Funktion, Beschreibung, Parameter):

cp_delete_session()	- Keine Beschreibung verfügbar. -	$variable
cp_set_session()	- Keine Beschreibung verfügbar. -	$variable, $value

6.9.22. Datei "/functions/cp.statistic.php"

Funktionen in der Datei "/functions/cp.statistic.php" (Name der Funktion, Beschreibung, Parameter):

cp_statistic()	- Keine Beschreibung verfügbar. -	
cp_statistic_browser()	- Keine Beschreibung verfügbar. -	$browser, $return_key = ""
cp_statistic_info()	- Keine Beschreibung verfügbar. -	
cp_statistic_isbot()	Überprüft, ob es sich bei einem Besucher um einen Bot handelt. Ist dies der Fall, dann kann eine bestimmte Aktion geblockt werden.	$user_agent
cp_statistic_javascript()	- Keine Beschreibung verfügbar. -	$browser
cp_statistic_language()	- Keine Beschreibung verfügbar. -	$language
cp_statistic_os()	- Keine Beschreibung verfügbar. -	$os

6.9.23. Datei "/functions/cp.string.php"

Funktionen in der Datei "/functions/cp.string.php" (Name der Funktion, Beschreibung, Parameter):

cp_is_number()	- Keine Beschreibung verfügbar. -	$string
cp_is_string()	- Keine Beschreibung verfügbar. -	$string
cp_string_calculate()	- Keine Beschreibung verfügbar. -	$string
cp_string_convert()	- Keine Beschreibung verfügbar. -	$array, $seperator = ","
cp_string_convert_unicode()	- Keine Beschreibung verfügbar. -	$string, $language = ""
cp_string_count_words	- Keine Beschreibung verfügbar. -	$string

()		
cp_string_decode()	- Keine Beschreibung verfügbar. -	$string, $level = "4"
cp_string_encode()	- Keine Beschreibung verfügbar. -	$string, $level = "4"
cp_string_random()	- Keine Beschreibung verfügbar. -	$length
cp_string_rm_spaces()	- Keine Beschreibung verfügbar. -	$string
cp_string_shorten()	- Keine Beschreibung verfügbar. -	$string, $max_length = "50", $points = "1"
cp_string_shorten_html ()	- Keine Beschreibung verfügbar. -	$string, $max_length = "50", $points = "1"
cp_string_spaces()	- Keine Beschreibung verfügbar. -	$string
cp_string_summarize()	- Keine Beschreibung verfügbar. -	$string, $words = "10"
cp_string_swap()	- Keine Beschreibung verfügbar. -	$var_a, $var_b
cp_string_unicode()	- Keine Beschreibung verfügbar. -	$string

6.9.24. Datei "/functions/cp.template.php"

Funktionen in der Datei "/functions/cp.template.php" (Name der Funktion, Beschreibung, Parameter):

cp_template_assign()	- Keine Beschreibung verfügbar. -	$string
cp_template_client()	- Keine Beschreibung verfügbar. -	$params, $content
cp_template_date()	- Keine Beschreibung verfügbar. -	$format, $time = ""
cp_template_dump()	- Keine Beschreibung verfügbar. -	$file, $string
cp_template_format()	- Keine Beschreibung verfügbar. -	$string, $template = ""
cp_template_language()	- Keine Beschreibung verfügbar. -	$module, $value
cp_template_open()	- Keine Beschreibung verfügbar. -	$template, $mode = "complete", $is_file = "1", $parse_container = "1", $template_options = array()
cp_template_output()	- Keine Beschreibung verfügbar. -	$content, $return = "", $clean_ignore = "1", $parse_php = "1", $no_replace = "0"
cp_template_process()	- Keine Beschreibung verfügbar. -	$content, $mode = "complete", $parse_container = "1", $template_options = array(), $template = ""
cp_template_replace()	- Keine Beschreibung verfügbar. -	$content, $var, $string
cp_template_rewrite()	- Keine Beschreibung verfügbar. -	$string
cp_template_rights()	- Keine Beschreibung verfügbar. -	$string, $content
cp_template_switch()	- Keine Beschreibung verfügbar. -	$type, $variable, $content
cp_template_template()	- Keine Beschreibung verfügbar. -	$string, $content
cp_template_texteffect ()	- Keine Beschreibung verfügbar. -	$string

6.9.25. Datei "/functions/cp.url.php"

Funktionen in der Datei "/functions/cp.url.php" (Name der Funktion, Beschreibung, Parameter):

cp_url_body()	- Keine Beschreibung verfügbar. -	$url, $mode = "0"
cp_url_clean()	- Keine Beschreibung verfügbar. -	$string
cp_url_connect()	- Keine Beschreibung verfügbar. -	$query
cp_url_element()	- Keine Beschreibung verfügbar. -	$query, $type = ""
cp_url_extend()	- Keine Beschreibung verfügbar. -	$query
cp_url_format()	- Keine Beschreibung verfügbar. -	$query
cp_url_frameset()	- Keine Beschreibung verfügbar. -	$url, $mode = "0"
cp_url_header()	- Keine Beschreibung verfügbar. -	$query, $mode = ""
cp_url_images()	- Keine Beschreibung verfügbar. -	$url, $mode = "0"
cp_url_is_host()	- Keine Beschreibung verfügbar. -	$url
cp_url_links()	- Keine Beschreibung verfügbar. -	$url, $mode = "0", $inside_links = "1"
cp_url_lookup()	- Keine Beschreibung verfügbar. -	$query
cp_url_metatags()	- Keine Beschreibung verfügbar. -	$url, $mode = ""
cp_url_port()	- Keine Beschreibung verfügbar. -	$host, $port, $max_time = "10"
cp_url_title()	- Keine Beschreibung verfügbar. -	$query

6.9.26. Datei "/functions/cp.user.php"

Funktionen in der Datei "/functions/cp.user.php" (Name der Funktion, Beschreibung, Parameter):

cp_get_userip()	Liefert die IP-Adresse des Benutzers zurück.	
cp_user_grouprights()	Liefert die Benutzerberechtigungen im Artikelmodul zurück, für ein bestimmtes Recht pro Kategorie, für die Kategorie allgemein oder insgesamt. Rückmeldung: "0" = Recht nicht vorhanden, "1" = Recht vorhanden.	$module, $number = "", $right = ""
cp_user_info()	Liefert eine bestimmte Information über einen Benutzer zurück.	$number, $field
cp_user_lastlogin()	Liefert das Datum des letzten Logins eines Benutzers zurück.	$date_format = ""
cp_user_rights()	Liefert die Benutzerberechtigungen für ein bestimmtes Recht zurück.	$right_name

6.9.27. Datei "/functions/cp.wddx.php"

Funktionen in der Datei "/functions/cp.wddx.php" (Name der Funktion, Beschreibung, Parameter):

cp_wddx_deserialize()	Deserialisiert WDDX-Datenpakete.	$wddx_string
cp_wddx_serialize()	Serialisiert WDDX-Datenpakete.	$wddx_sql, $wddx_title = ""

6.9.28. Datei "/functions/cp.xml.php"

Funktionen in der Datei "/functions/cp.xml.php" (Name der Funktion, Beschreibung, Parameter):

add_attribute()	- Keine Beschreibung verfügbar. -	$node, $name = "", $value = ""
add_content()	- Keine Beschreibung verfügbar. -	$node, $content
add_node()	- Keine Beschreibung verfügbar. -	$base_node, $new_node
convert()	- Keine Beschreibung verfügbar. -	$string
create()	- Keine Beschreibung verfügbar. -	
db_input()	- Keine Beschreibung verfügbar. -	$file, $table, $root_tag
db_output()	- Keine Beschreibung verfügbar. -	$file, $db_table, $root_tag = "db_table"
get_format()	- Keine Beschreibung verfügbar. -	$array, $level = "0", $old_key = "", $options = ""
get_path()	- Keine Beschreibung verfügbar. -	$node
get_structure()	- Keine Beschreibung verfügbar. -	$vals, &$i
open()	- Keine Beschreibung verfügbar. -	$file, $cache_time = "3600"
remove_attribute()	- Keine Beschreibung verfügbar. -	$node
remove_content()	- Keine Beschreibung verfügbar. -	$base_node
remove_node()	- Keine Beschreibung verfügbar. -	$base_node
remove_sub_nodes()	- Keine Beschreibung verfügbar. -	$base_node, $sub_node
save()	- Keine Beschreibung verfügbar. -	$file, $format = "xml", $options = ""
save_php()	- Keine Beschreibung verfügbar. -	$array, $level = "0", $old_key = "", $prefix = "array", $smart_tag = ""
save_xml()	- Keine Beschreibung verfügbar. -	$options
set_content()	- Keine Beschreibung verfügbar. -	$node, $content
set_input()	- Keine Beschreibung verfügbar. -	$content
set_special()	- Keine Beschreibung verfügbar. -	$array
show()	- Keine Beschreibung verfügbar. -	$return = ""
show_cdata()	- Keine Beschreibung verfügbar. -	$content
show_feed()	- Keine Beschreibung verfügbar. -	$url, $template = "", $limit = "10", $return = "1"

6.9.29. Datei "/functions/cp.zip.php"

Funktionen in der Datei "/functions/cp.zip.php" (Name der Funktion, Beschreibung, Parameter):

add_dir()	- Keine Beschreibung verfügbar. -	$dir
add_file()	- Keine Beschreibung verfügbar. -	$file, $zip_name = "", $zip_dir = ""
close()	- Keine Beschreibung verfügbar. -	
download()	- Keine Beschreibung verfügbar. -	$file, $content
get_data()	- Keine Beschreibung verfügbar. -	$file
get_dos_time()	- Keine Beschreibung verfügbar. -	$time
get_file_position()	- Keine Beschreibung verfügbar. -	$file
get_header()	- Keine Beschreibung verfügbar. -	
load_file()	- Keine Beschreibung verfügbar. -	
open()	- Keine Beschreibung verfügbar. -	$file
pack()	- Keine Beschreibung verfügbar. -	$file, $overwrite = "1", $download = "0"
show_content()	- Keine Beschreibung verfügbar. -	$file = ""
show_info()	- Keine Beschreibung verfügbar. -	
unpack()	- Keine Beschreibung verfügbar. -	$source, $file, $dir, $uncompress = "1"

6.9.30. Datei "/modules/ads/class.admin.php"

Funktionen in der Datei "/modules/ads/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_ads()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_banner()	Generiert das Formular zum Hinzufügen eines neuen Banners.	
do_add_banner_2()	Fügt einen neuen Banner in die Datenbank ein.	$form
do_add_client()	Fügt einen neuen Kunden in die Datenbank ein.	$form
do_choose_ads()	Ermöglicht eine gleichzeitige Änderung mehrerer Werbemittel (z.B. Aktivierung, Deaktivierung).	$action, $checkbox
do_delete_banner()	Darstellung der Bestätigungsseite zum Löschen eines Werbebanners.	$number
do_delete_banner_2()	Löscht einen vorhandenen Banner aus der Datenbank.	$form, $redirect = "1"
do_delete_client()	Löscht einen vorhandenen Kunden aus der Datenbank.	$number
do_edit_banner()	Generiert das Formular zur Bearbeitung eines vorhandenen Banners.	$number
do_edit_banner_2()	Aktualisiert die Daten zu einem vorhandenen Banner.	$form
do_edit_client()	Generiert das Formular zur Bearbeitung eines vorhandenen Kunden.	$number
do_edit_client_2()	Aktualisiert die Daten zu einem bestehenden Kunden.	$form
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$form
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.31. Datei "/modules/ads/class.base.php"

Funktionen in der Datei "/modules/ads/class.base.php" (Name der Funktion, Beschreibung, Parameter):

do_menu()	contentpapst_ads::do_menu Generiert die Auswahl für die verfügbaren Typen von Werbung.	$active, $mode = "", $get_array = "'
do_show()	Aktualisiert die Klickstatistik für einen Banner und leitet zur URL weiter.	$clicked

6.9.32. Datei "/modules/ads/class.index.php"

Funktionen in der Datei "/modules/ads/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_ads()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_client()	Generierung des Formulares zur Bearbeitung der Kundendaten.	$form
do_edit_client()	Aktualisiert die Daten eines Kunden in der Datenbank.	$form
do_login_now()	Nimmt den Login des Benutzers vor bzw. leitet auf die eigentliche Login-Funktion weiter.	$form
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$form
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.33. Datei "/modules/ads/class.webservices.php"

Funktionen in der Datei "/modules/ads/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.34. Datei "/modules/ads/container.php"

Funktionen in der Datei "/modules/ads/container.php" (Name der Funktion, Beschreibung, Parameter):

ads_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Für das "ads"-Modul ist dies die Darstellung der einzelnen Banner etc.. Aufruf in den Vorlagen: {cp:container module="ads" [type="1"]}	$forn

6.9.35. Datei "/modules/article/class.admin.php"

Funktionen in der Datei "/modules/article/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_article ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_article()	Zeigt das Formular zum Hinzufügen eines neuen Artikels an.	
do_add_article_2()	Fügt der Datenbank einen neuen Artikel hinzu.	$form, $articledesc, $articletext
do_add_category()	Erzeugt das Formular zum Hinzufügen neuer Kategorien.	$form
do_add_category_2()	Fügt die neuen Kategorien in die Datenbank ein.	$form
do_archive_articles()	Legt alle Artikel einer Kategorie im Archiv ab.	$number
do_article_now()	Leitet von einer beliebigen Seite auf die Bearbeitung eines Artikels weiter.	$number
do_blocked_contact()	Erzeugt ein Formular zur Kontaktaufnahme bei einem blockierten Artikel.	$number
do_blocked_contact_2()	Versendet die Daten aus dem Formular zur Kontaktaufnahme.	$form
do_category()	Generiert die Darstellung für die Kategorieübersicht.	
do_change_status()	Ändert den Status eines Artikels auf verfügbar / geschützt.	$number
do_choose_articles()	Ermöglicht eine gleichzeitige Änderung mehrerer Artikel (z.B. Aktivierung, Deaktivierung).	$action, $checkbox
do_comment()	Baut das Formular zum Verfassen eines Kommentares auf.	$number
do_comment_2()	Speichert den Kommentar für den aktuellen Artikel ab.	$form
do_delete_article()	Löscht einen vorhandenen Artikel aus der Datenbank.	$number, $redirect = "1"

do_delete_category()	Löscht abhängig von den Berechtigungen vorhandene Kategorien.	$number
do_duplicate_article()	Kopiert einen bereits vorhandenen Artikel inklusive der Angaben zur Sichtbarkeit.	$number, $redirect = "1"
do_edit_article()	Erzeugt das Formular zum Bearbeiten eines vorhandenen Artikels.	$number
do_edit_article_2()	Aktualisiert die Angaben zu einem Artikel in der Datenbank.	$form, $articledesc, $articletext
do_edit_article_text()	Zeigt ein Formular für die Bearbeitung eines Artikels als normalen Text an.	$number
do_edit_article_text_2()	Aktualisiert die Angaben zu einem Artikel in der Datenbank.	$form
do_edit_category()	Erzeugt das Formular zur Bearbeitung von Kategorien.	$number
do_edit_category_2()	Aktualisiert die Daten zu einer Kategorie in der Datenbank.	$form
do_error()	Zeigt eine Fehlermeldung zu einem Artikel an, wenn dieser blockiert ist.	$number
do_frontend_edit()	Aktualisiert einen über das Frontend bearbeiteten Artikel in der Datenbank.	$form
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$form
do_main_results()	Holt die zutreffenden Artikel aus der Datenbank.	$form
do_quickstart()	Ermöglich Konfiguration und Start der Quickstart-Funktion.	$form
do_quickstart_now()	Löst die Quickstart-Funktion aus und erstellt Artikel, Artikelkategorien und Navigationspunkte.	$form
do_search_now()	Leitet vom Suchformular auf die eigentliche Suchergebnisseite weiter.	$form
do_section_menu()	Erstellt das Dropdown-Menü mit der Hierarchie der Kategorien.	$level, $count = "0", $selection = "" $rights_mode = ""
do_shortnavigation()	Zeigt die Kurznavigation auf der linken Seite an.	
do_show_list()	Liefert die Anzeige der vorhandenen Artikel.	$row, $actions
do_status_info()	Erzeugt eine neue Aufgabe für den Artikel bei zu geringen Berechtigungen.	$number
do_structure()	Zeigt die Struktur der Kategorien und Artikel an.	$number
do_structure_path()	Zeigt die Breadcrumb-Navigation für die Kategorien an.	$number, $level = "0"
do_structure_show()	Generiert die Baumstruktur mit den einzelnen Kategorien und Artikeln.	$number, $count = "0", $actions
do_textedit()	Aktiviert oder deaktiviert den Texteditor dauerhaft für einen bestimmten Artikel.	$number, $type
do_view_list()	Liefert den Feldinhalt für die Sichtbarkeit von Artikeln für die einzelnen Benutzergruppen und (sofern vorhanden) die Kundengruppen von CP::Shop zurück.	$selected = array(), $type = "1"
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.36. Datei "/modules/article/class.base.php"

Funktionen in der Datei "/modules/article/class.base.php" (Name der Funktion, Beschreibung, Parameter):

create_javascript()	Erzeugt den Javascript-Code zur Überprüfung der Formulare.	$force_textarea = "0", $force_status = "0", $form_name = "form_standard", $frontend_editor = "0"
create_wysiwyg_options ()	Erzeugt die allgemeingültigen Einstellungen für den WYSIWYG-Editor.	
do_clean_html()	Schließt in einem gekürzten String eventuell noch offene HTML-Tags.	$string
do_menu()	Gibt ein Array mit allen Kategorien inkl. der Unterkategorien zurück.	$level, $count = "0", $show_permalinks = "0", $active = ""
do_related()	Liefert anhand der Schlüsselwörter die verwandten Artikel zurück.	$number
do_show()	Formatiert einen einzelnen Artikel und verarbeitet die darin enthaltenen {cp:tags}.	$form
do_views()	Überprüft ob ein Zugriff auf den Artikel je nach Benutzergruppe möglich ist.	$number, $return = "0"
generate_file_path()	Erzeugt zentral die Pfade zu den Beiträgen, zu den Kategorien etc. innerhalb der PHP-Dateien.	$type, $array = array(), $full = "0"

6.9.37. Datei "/modules/article/class.index.php"

Funktionen in der Datei "/modules/article/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_article ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_entry()	Generiert die Einzelseite für die Darstellung eines Artikels.	$number, $page
do_entry_path()	Erzeugt die Breadcrumb-Navigation für die Artikel/Einträge.	$number, $entry
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	
do_navigation()	Erzeugt die Navigation, welche über "{cp:level}" eingebunden wird.	$nav_array, $level
do_section()	Erzeugt die Einzelansicht einer Kategorie.	$number, $page
do_section_path()	Erzeugt die Breadcrumb-Navigation für die Kategorien.	$number
do_show_entry()	Zeigt einen fertig formatierten und verarbeiteten Eintrag an.	$number, $page, $is_status = "", $sec_number = "", $pre_content = ""
do_vote()	Speichert eine einzelne Stimme, wenn für den Artikel abgestimmt wurde.	$number, $form
frontend_editing()	Behandelt die Einblendung der Frontend-Editing-Leiste für berechtigte Benutzer.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.38. Datei "/modules/article/class.webservices.php"

Funktionen in der Datei "/modules/article/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

do_articles()	Gibt Daten zu Artikeln anhand eines SQL-Queries als XML-Datei zurück. Aufruf: http://www.domain.de/pfad/webservices.php?file=article&number=123&(format)=…	$form, $query, $article_class
do_rss()	Liefert die letzten Einträge als RSS 2.0-Feed aus. Aufruf: http://www.domain.de/pfad/webservices.php?file=article&type=rss&(limit|start|section)=…	$form, $article_class
process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.39. Datei "/modules/article/container.php"

Funktionen in der Datei "/modules/article/container.php" (Name der Funktion, Beschreibung, Parameter):

article_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Für die Artikel sind dies Container für die Detail- und Listenansicht sowie für die Darstellung der verwandten Artikel. Aufruf in den Vorlagen: {cp:container module="article" […]}	$form
article_container_subcategories ()	Liefert ein Array der Unterkategorien einer Kategorie zurück.	$number = "0"

6.9.40. Datei "/modules/backup/class.admin.php"

Funktionen in der Datei "/modules/backup/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_backup ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_entry()	Formular zum Hinzufügen eines neuen Eintrages.	$table
do_add_entry_2()	Fügt den neuen Eintrag in die Datenbank ein.	$form
do_add_fields()	Formular zum Hinzufügen neuer Felder zu einer Datenbanktabelle.	$form
do_add_fields_now()	Fügt ein neues Feld in eine Datenbanktabelle ein.	$form
do_clean()	Löscht alle Einträge in einer vorhandenen Datenbanktabelle.	$table, $action
do_clean_table()	Löscht alle Datensätze aus den angegebenen Datenbanktabellen.	$select, $table
do_create_db()	Legt eine neue, vorerst leere Datenbank an.	$form
do_csv_export()	Ermöglicht den Export von Datenbankinhalten im CSV-Format.	$table, $options = array()
do_delete_entry()	Reinigt die URL vor dem Löschen von vorhandenen Einträgen.	$query
do_delete_field()	Löscht ein vorhandenes Feld aus einer Datenbanktabelle.	$table, $field
do_delete_index()	Löscht einen vorhandenen Index aus einer Datenbanktabelle.	$table, $index
do_delete_query()	Setzt anhand der übergebenen Variablen einen SQL-Befehl zum Löschen zusammen.	$table, $keys, $data
do_drop_db()	Löscht eine komplette Datenbank samt aller Tabellen und Inhalte.	$form
do_drop_table()	Löscht eine einzelne Datenbanktabelle.	$table
do_edit_db()	Statusabfrage, ob eine Datenbank wirklich gelöscht werden soll.	$form
do_edit_entry()	Formular zur Bearbeitung eines vorhandenen Eintrages.	$table, $query
do_edit_entry_2()	Aktualisierung eines Eintrages in der Datenbank.	$form
do_edit_entry_inplace()	Speichert die per In-Place-Editor geänderten Werte ab.	$form
do_edit_field()	Formular zur Bearbeitung eines Feldes einer Datenbanktabelle.	$table, $field
do_edit_field_now()	Format eines Feldes einer Datenbanktabelle wird in der Datenbank aktualisiert.	$form
do_edit_query()	Setzt anhand der übergebenen Variablen einen SQL-Befehl zum Abfragen zusammen.	$table, $keys, $data
do_export()	Formular für den erweiterten Export von Inhalten in verschiedenen Formaten.	$tables = ""
do_export_now()	Nimmt den Export der Datenbankinhalte und -tabellen in verschiedenen Formaten vor.	$form
do_format_date()	Wandelt ein Datum in das für die Datenbank benötigte Format um.	$date
do_get_presets()	Liefert ein Array mit grundlegenden Datenbankeinstellungen zurück.	
do_import()	Formular für den Import von SQL-Befehlen.	$form
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$form
do_optimize()	Optimiert einzelne bzw. wahlweise alle Datenbanktabellen in einer Datenbank.	$select, $table
do_out()	Leitet auf eine externe Seite weiter. Brücken-Funktion, damit die genaue URL aus der Datenbank nicht als Referer von der Zielseite ermittelt werden kann.	$url
do_quick_backup()	Generiert mit nur einem Klick ein Backup der gesamten Datenbank für diesen Mandanten und exportiert diese.	$form
do_repair()	Repariert korrupte Datensätze in einzelnen bzw. mehreren Datenbanktabellen.	$select, $table
do_router()	Leitet an die einzelnen Funktionen für die Tabellenbearbeitung weiter.	$setaction, $checkbox
do_search()	Zeigt das Formular für die Volltextsuche in der Datenbank an.	$form
do_set_inplace()	Aktiviert oder deaktiviert das In-Place-Editing für die Datenbanktabellen.	$type
do_show_tables()	Anzeige der einzelnen Einträge in einer Datenbanktabelle.	$table, $perpage, $limited, $set_limit, $selected, $page, $sorder, $dir, $show_options, $lastquery
do_show_view()	Leitet auf die Ansicht der Datenbanktabellen weiter.	$form, $select_tables = array()
do_table_comment()	Fügt einen Kommentar für eine Datenbanktabelle hinzu.	$form

do_table_extended()	Setzt die erweiterten Eigenschaften für eine Datenbanktabelle.	$form
do_table_order()	Legt die Standardsortierung für eine Datenbanktabelle fest.	$form
do_table_rename()	Weist einer Datenbanktabelle einen neuen Namen zu.	$form
do_table_router()	Leitet an die einzelnen Funktionen für die Tabellenbearbeitung weiter.	$table, $setaction, $checkbox
do_table_structure()	Zeigt Informationen über die Struktur einer Datenbanktabelle an.	$table
do_table_type()	Ändert den Tabellentyp einer vorhandenen Datenbanktabelle.	$form
do_table_type_array()	Liefert ein Array über die möglichen Tabellentypen zurück.	
do_update_db()	Importiert neue Inhalte als SQL in die Datenbank.	$form
do_view_table()	Formular für die partielle Anzeige von Tabelleninhalten.	$table, $form, $page
do_xml_export()	Ermöglicht den Export von Datenbankinhalten im XML-Format.	$table, $options = array()
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.41. Datei "/modules/backup/class.base.php"

Funktionen in der Datei "/modules/backup/class.base.php" (Name der Funktion, Beschreibung, Parameter):

do_backup()	Führt das Backup von Datenbanktabellen und/oder Inhalten durch.	$checkbox, $setaction, $return = "0", $showsite = "0", $quer = ""
do_structure()	Liefert die Struktur einer Datenbanktabelle als SQL für ein Backup zurück.	$table, $drop_table = "0"
do_table_info()	Liefert Informationen zu einer Datenbanktabelle zurück.	$table
do_tables()	Liefert den Inhalt einer Datenbanktabelle als SQL für ein Backup zurück.	$table, $showsite = "0", $query = ""

6.9.42. Datei "/modules/backup/class.index.php"

Funktionen in der Datei "/modules/backup/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_backup ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.43. Datei "/modules/backup/class.webservices.php"

Funktionen in der Datei "/modules/backup/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.44. Datei "/modules/backup/container.php"

Funktionen in der Datei "/modules/backup/container.php" (Name der Funktion, Beschreibung, Parameter):

backup_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="backup" [...]}	$forn

6.9.45. Datei "/modules/editor/admin.php"

Funktionen in der Datei "/modules/editor/admin.php" (Name der Funktion, Beschreibung, Parameter):

editor_colors()	Regelt die Anzeige der Auswahl für Vorder- und Hintergrundfarbe.	
editor_enlarge()	Regelt den Zugriff auf den erweiterten Editor.	$type
editor_images()	Regelt die Anzeige der Auswahl für Sonderzeichen.	$path, $userfile = ""
editor_mode()	Regelt den Zugriff auf einzelne Funktionen des Editors.	$mode, $form
editor_specialchars()	Regelt die Anzeige der Auswahl für Sonderzeichen.	

6.9.46. Datei "/modules/editor/enlarge.php"

Funktionen in der Datei "/modules/editor/enlarge.php" (Name der Funktion, Beschreibung, Parameter):

editor_enlarge_main()	Zeigt den erweiterten WYSIWYG-Editor mit umfangreichen Funktionen an.	$type
editor_enlarge_start()	Start-Funktion vor dem Öffnen des erweiterten WYSIWYG-Editors.	

6.9.47. Datei "/modules/editor/index.php"

Funktionen in der Datei "/modules/editor/index.php" (Name der Funktion, Beschreibung, Parameter):

editor_cache()	Lädt einen vorhadenen Inhalt / eine Datei in den Editor.	$object
editor_colors()	Regelt die Anzeige der Auswahl für Vorder- und Hintergrundfarbe.	
editor_enlarge()	Regelt den Zugriff auf den erweiterten Editor.	$type, $content
editor_images()	Regelt die Anzeige der Auswahl für Sonderzeichen.	$path, $userfile = ""
editor_main()	Ruft die Hauptansicht des erweiterten Editors auf.	
editor_mode()	Regelt den Zugriff auf einzelne Funktionen des Editors.	$mode, $form
editor_preview()	Regelt die Anzeige des Vorschau-Fensters.	$type, $content
editor_preview_now()	Zeigt die Vorschau des Inhalts auf der Basis einer Vorlage an.	$form
editor_specialchars()	Regelt die Anzeige der Auswahl für Sonderzeichen.	

6.9.48. Datei "/modules/editor/wysiwyg.anchor.php"

Funktionen in der Datei "/modules/editor/wysiwyg.anchor.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_anchor()	Liefert die Funktionalität für das Einfügen von Textmarken.	$form

6.9.49. Datei "/modules/editor/wysiwyg.button.php"

Funktionen in der Datei "/modules/editor/wysiwyg.button.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_button()	Liefert die Funktionalität für die Schaltflächen.	$form

6.9.50. Datei "/modules/editor/wysiwyg.checkbox.php"

Funktionen in der Datei "/modules/editor/wysiwyg.checkbox.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_checkbox()	Liefert die Funktionalität für die Kontrollkästchen.	$form

6.9.51. Datei "/modules/editor/wysiwyg.dropdown.php"

Funktionen in der Datei "/modules/editor/wysiwyg.dropdown.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_dropdown()	Liefert die Funktionalität für das Einfügen von Dropdownfeldern.	$form

6.9.52. Datei "/modules/editor/wysiwyg.email.php"

Funktionen in der Datei "/modules/editor/wysiwyg.email.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_email()	Liefert die Funktionalität für das Einfügen von Email-Adressen.	$form

6.9.53. Datei "/modules/editor/wysiwyg.find.php"

Funktionen in der Datei "/modules/editor/wysiwyg.find.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_find()	Liefert die Funktionalität für die Suche innerhalb des Editors.	$form

6.9.54. Datei "/modules/editor/wysiwyg.form.php"

Funktionen in der Datei "/modules/editor/wysiwyg.form.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_form()	Liefert die Funktionalität für ein Formular.	$form

6.9.55. Datei "/modules/editor/wysiwyg.hidden.php"

Funktionen in der Datei "/modules/editor/wysiwyg.hidden.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_hidden()	Liefert die Funktionalität für ein verstecktes Formularfeld.	$form

6.9.56. Datei "/modules/editor/wysiwyg.images.php"

Funktionen in der Datei "/modules/editor/wysiwyg.images.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_images()	Liefert die Funktionalität für das Einfügen von Bildern.	$path = "", $userfile = ""

6.9.57. Datei "/modules/editor/wysiwyg.link.php"

Funktionen in der Datei "/modules/editor/wysiwyg.link.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_link()	Liefert die Funktionalität für das Einfügen von Links.	$form

6.9.58. Datei "/modules/editor/wysiwyg.radio.php"

Funktionen in der Datei "/modules/editor/wysiwyg.radio.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_radio()	Liefert die Funktionalität für Optionsfelder.	$form

6.9.59. Datei "/modules/editor/wysiwyg.specialchars.php"

Funktionen in der Datei "/modules/editor/wysiwyg.specialchars.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_specialchars()	Liefert die Funktionalität für die Sonderzeichen.	

6.9.60. Datei "/modules/editor/wysiwyg.table.php"

Funktionen in der Datei "/modules/editor/wysiwyg.table.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_table()	Liefert die Funktionalität für das Einfügen von Tabellen.	$form

6.9.61. Datei "/modules/editor/wysiwyg.textarea.php"

Funktionen in der Datei "/modules/editor/wysiwyg.textarea.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_textarea()	Liefert die Funktionalität für einen Textbereich.	$form

6.9.62. Datei "/modules/editor/wysiwyg.textfield.php"

Funktionen in der Datei "/modules/editor/wysiwyg.textfield.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_textfield()	Liefert die Funktionalität für Textfelder.	$form

6.9.63. Datei "/modules/editor/wysiwyg.wizard.php"

Funktionen in der Datei "/modules/editor/wysiwyg.wizard.php" (Name der Funktion, Beschreibung, Parameter):

wysiwyg_wizard()	Liefert die Funktionalität für den {cp:tag}-Wizard.	$form
wysiwyg_wizard_title()	Formatiert die Anzeige innerhalb des {cp:tag}-Wizards.	$string, $content = ""

6.9.64. Datei "/modules/fast_gallery/class.admin.php"

Funktionen in der Datei "/modules/fast_gallery/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_fast_gallery ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.65. Datei "/modules/fast_gallery/class.index.php"

Funktionen in der Datei "/modules/fast_gallery/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_fast_gallery ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_main()	Darstellung eines einzelnen Bildes in der Originalgröße.	$show
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.66. Datei "/modules/fast_gallery/class.webservices.php"

Funktionen in der Datei "/modules/fast_gallery/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.67. Datei "/modules/fast_gallery/container.php"

Funktionen in der Datei "/modules/fast_gallery/container.php" (Name der Funktion, Beschreibung, Parameter):

fast_gallery_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Für das "fast_gallery"-Modul ist dies die Darstellung der Galerie. Aufruf: {cp:container module="fast_gallery" [path="grafiken"]} etc.	$form

6.9.68. Datei "/modules/forms/class.admin.php"

Funktionen in der Datei "/modules/forms/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_forms ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_form()	Formular zum Hinzufügen eines neuen Modules.	$form
do_add_form_template ()	Formular zum Hinzufügen einer Modulvorlage.	$form
do_change_position()	Ändert die Position von Feldern in einem vorhandenen Modul.	$form
do_check_templates()	Überprüft, ob alle Vorlagen vorhanden sind und legt fehlende Vorlagen an.	
do_clean_database()	Ermittelt nicht mehr benötigte Datensätze, Felder, Zugriffsberechtigungen etc.	
do_collection()	Hauptansicht für eine einzelne Collection.	$form
do_comment()	Eine Notiz zu einem Eintrag eines Modules verfassen.	$form
do_comment_now()	Notiz wird hinzugefügt oder gelöscht.	$form
do_convert_database()	Lagert die Einträge eines datenbankbasierten Modules in eine eigene Datenbanktabelle aus.	$form
do_delete_all()	Alle Einträge inklusive aller Dateien etc. in einem Modul werden gelöscht.	$form
do_delete_collection()	Löscht eine Collection, sofern keine Module zugeordnet sind.	$form
do_delete_dataset()	Ein einzelner Eintrag in einem Modul wird gelöscht.	$form
do_delete_field()	Löscht ein Feld samt aller Daten aus einem vorhandenen Modul.	$form
do_delete_file()	Eine Datei aus einem Feld eines Modules wird gelöscht.	$form
do_delete_form()	Ein komplettes Modul wird inklusive aller Dateien, Vorlagen etc. gelöscht.	$form
do_delete_template()	Eine Modulvorlage wird aus dem System entfernt.	$form
do_download_template ()	Eine Modulvorlage wird als Download vorbereitet.	$form
do_duplicate_dataset()	Ein einzelner Eintrag in einem Modul wird dupliziert.	$form
do_edit_form()	Formular zum Bearbeiten eines vorhandenen Modules.	$form
do_error()	Ausgabe verschiedener Fehlermeldungen.	$type, $value
do_form_collection()	Eine Collection anlegen oder bearbeiten.	$form
do_form_collection_2()	Verarbeitung der Daten zum Hinzufügen und Bearbeiten von Collections.	$form
do_form_form()	Generierung des Formulares zum Hinzufügen / Bearbeiten von Modulen (Seite 1).	$form
do_formrights()	Zeigt die Übersicht der Rechte zu einem Modul an.	$form
do_generate_form_fields ()	Verarbeitung der Daten aus dem Formular zum Hinzufügen / Bearbeiten von Modulen (Seite 1).	$form
do_generate_form_fields_2 ()	Generierung des Formulares zum Hinzufügen / Bearbeiten von Modulen (Seite 2).	$form
do_generate_form_fields_3 ()	Abschließende Zusammenfassung der Daten und Generierung des Modules.	$form
do_generate_template()	Generierung des Modules aus einer Modulvorlage heraus.	$form
do_import_module()	Eine neue Modulvorlage wird importiert.	$form
do_insert_dataset()	Anzeige des Formulares zum Hinzufügen eines Eintrages.	$form
do_insert_dataset_now ()	Ein neuer Eintrag wird inklusive aller Felder zum Modul hinzugefügt.	$form
do_livesearch()	Ermittelt verfügbare Werte für die Auto-Vervollständigung von einzelnen Feldern vom Typ "Textfeld".	$form
do_main()	Hauptansicht mit der Auswahl der verfügbaren Module und Collections.	
do_message()	Gibt eine Fehlermeldung auf einer neuen Seite aus.	$form
do_metaedit()	Ermöglicht die gleichzeitige Bearbeitung mehrerer Einträge (u.a. Aktivierung, Deaktivierung, Duplizierung).	$form, $action, $checkbox
do_modify_field()	Formular zum Hinzufügen / Bearbeiten eines Feldes.	$form
do_move_field()	Verschiebt ein Feld eines CP::Forms-Modules um jeweils eine Position nach oben oder unten.	$form
do_prosearch()	Wechselt zwischen der einfachen und der erweiterten Suche in einem datenbankbasierten CP::Forms-Modul.	$form
do_refresh_template()	Generiert eine einzelne Vorlage entsprechend des aktuellen Aufbaus dieses Modules.	$form
do_save_template()	Speichert ein Modul als Modulvorlage im XML-Format ab.	$form
do_save_template_2()	Ermöglicht den Wechsel zur neuen Modulvorlage.	$form
do_set_formrights()	Setzt die einzelnen Rechte zu einem Modul.	$form
do_shortnavigation()	Zeigt die Kurznavigation auf der linken Seite an.	
do_show_collection()	Zeigt eine einzelne Collection an.	$form, $type, $show_link = "0"
do_show_fields()	Zeigt die Auflistung der aktuell in einem Modul vorhandenen Felder an.	$from = array()
do_show_now()	Umleitung für die Anzeige eines Suchergebnisses in einem Modul.	$form = array(), $prosearch = array()
do_show_taginfo()	An das Modul angepasste Erklärung der zur Verfügung stehenden {cp:tags}.	$form
do_show_templateinfo()	An das Modul angepasste Erklärung der zur Verfügung stehenden Vorlagen.	$form
do_statistic()	Zeigt eine statistische Auswertung der in den Einträgen eines datenbankbasierten Modules ausgewählten Werte an.	$form
do_templates()	Übersichtsseite der Modulvorlagen.	$form

do_update_dataset()	Anzeige des Formulares zum Bearbeiten eines Eintrages.	$form
do_update_dataset_now ()	Ein Eintrag eines Modules wird inklusive aller Felder aktualisiert.	$form
do_update_field()	Fügt ein Feld zu einem Modul hinzu bzw. aktualisiert die Angaben eines bestehenden Feldes.	$form
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.69. Datei "/modules/forms/class.base.php"

Funktionen in der Datei "/modules/forms/class.base.php" (Name der Funktion, Beschreibung, Parameter):

create_module()	Erzeugt ein neues bzw. aktualisiert ein bestehendes Modul inkl. aller Dateien.	$form, $is_update = "0"
create_template()	Erzeugt die unterschiedlichen Vorlagen für ein Modul.	$type, $form
do_dataset_form()	Generierung des Formulares zum Hinzufügen / Bearbeiten von Einträgen.	$form
do_export_datasets_now ()	Nimmt den Export von Einträgen für ein Modul vor.	$form
do_import_datasets_now ()	Nimmt den Import von Einträgen für ein Modul vor.	$form
do_import_export()	Import/Export von Einträgen für die Module.	$form
do_show()	Übersichtsseite eines Modules mit Suchfunktionen.	$form
do_show_content()	Stellt die passenden Einträge anhand der übergebenen Suchparameter zusammen.	$form, $this_options
do_show_status()	Zeigt die Erklärung für den Status der Einträge in der Übersicht an.	
form_check_files()	Überprüft, ob das Modul Felder für Dateien beinhaltet.	$module
get_clean_dataset()	Stellt zu einem Modul einen leeren Eintrag als assoziatives Array zusammen.	$module
get_clean_search_array ()	Ermittelt die in allen Abfragen passenden Einträge.	$array
get_client_validation()	Setzt die für eine clientseitige Validierung per Javascript benötigten Funktionen zusammen.	$module, $form, $return = "0", $add_slashes = "0", $show_status = "1"
get_dataset()	Stellt zu einem Eintrag alle Daten als assoziatives Array zusammen.	$module, $number, $only_active = "0", $selected_fields_array = array(), $blocked_fields_array = array()
get_field_number()	Ermittelt die Feldnummer eines bestimmten Feldes.	$module, $field
get_fields()	Stellt die Felder und Typen zu einem Modul zusammen.	$module, $order = ""
get_filetypes()	Ermittelt die für Import/Export verfügbaren Dateitypen.	
get_form()	Holt nach dem Ordner-Namen eines Modules die zugehörigen Angaben aus der Datenbank.	$module, $field = ""
get_form_number()	Holt nach dem Ordner-Namen eines Modules die zugehörige Nummer aus der Datenbank.	$module
get_info()	Holt die Informationen zu einem Eintrag der CP::Forms-Datenbanktabellen ab.	$type, $number, $field = ""
get_result_array()	Ermittelt die zutreffenden Einträge in der Datenbank für die Suche.	$query
get_user_rights()	Ermittelt die Rechte eines Benutzers für ein Modul oder ein einzelnes Feld.	$type, $module, $number, $right = ""
insert_dataset()	Fügt einen neuen Eintrag in die Datenbank ein.	$form
send_csvmail()	Erstellt die CSV-Datei für den Versand per Email im Anhang.	$form
send_xmlmail()	Erstellt die XML-Datei für den Versand per Email im Anhang.	$form
show_container()	Formatiert einen Eintrag für die Ausgabe als Container.	$module, $number, $form
update_dataset()	Aktualisiert einen vorhandenen Eintrag eines Modules.	$form

6.9.70. Datei "/modules/forms/class.index.php"

Funktionen in der Datei "/modules/forms/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_forms ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_email_dataset()	Versendet einen Eintrag per Email.	$form
do_insert_dataset()	Fügt einen neuen Eintrag zu einem Modul hinzu.	$form
do_output_datatype()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist (z.B. bei "Captchas").	$form
do_update_dataset()	Aktualisiert einen vorhandenen Eintrag in einem Modul.	$form
get_method()	Regelt den öffentlichen Zugriff auf die verfügbaren Methoden.	$type, $form
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.71. Datei "/modules/forms/class.webservices.php"

Funktionen in der Datei "/modules/forms/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

do_details()	Darstellung einzelner Einträge eines Modules.	$form
do_list()	Darstellung einer Liste von Einträgen.	$form
do_rssfeed()	Generiert einen RSS-Feed der Einträge eines Modules.	$form
process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.72. Datei "/modules/forms/container/details.php"

Funktionen in der Datei "/modules/forms/container/details.php" (Name der Funktion, Beschreibung, Parameter):

forms_container_details ()	Container für die Auflistung eines einzelnen Eintrages. Aufruf in den Vorlagen: {cp:container module="[MODULNAME]" type="details" number="[ID]" [template="vorlage.html"] [format="html"]}	$form
forms_container_details_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
forms_container_details_info ()	Liefert den Titel des Containers zurück.	

6.9.73. Datei "/modules/forms/container/form.php"

Funktionen in der Datei "/modules/forms/container/form.php" (Name der Funktion, Beschreibung, Parameter):

forms_container_form()	Container für die Anzeige eines Formulares zum Einfügen von Einträgen. Aufruf in den Vorlagen: {cp:container module="[MODULNAME]" type="form" [return="index.php?id=13"]}	$form
forms_container_form_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
forms_container_form_info ()	Liefert den Titel des Containers zurück.	

6.9.74. Datei "/modules/forms/container/list.php"

Funktionen in der Datei "/modules/forms/container/list.php" (Name der Funktion, Beschreibung, Parameter):

forms_container_list()	Container für die Auflistung mehrerer Einträge als Liste. Aufruf in den Vorlagen: {cp:container module="[MODULNAME]" type="list" [page="1"] [limit="10"] [order="number"] [dir="asc"] [template="vorlage.html"] [format="html"]}	$form
forms_container_list_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
forms_container_list_info ()	Liefert den Titel des Containers zurück.	

6.9.75. Datei "/modules/forms/container/pages.php"

Funktionen in der Datei "/modules/forms/container/pages.php" (Name der Funktion, Beschreibung, Parameter):

forms_container_pages ()	Container für die Erstellung einer Navigation für die Einträge zum Umblättern. Aufruf in den Vorlagen: {cp:container module="[MODULNAME]" type="pages" [page="1"] [per_page="10"]}	$form
forms_container_pages_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
forms_container_pages_info ()	Liefert den Titel des Containers zurück.	

6.9.76. Datei "/modules/forms/container/prosearch.php"

Funktionen in der Datei "/modules/forms/container/prosearch.php" (Name der Funktion, Beschreibung, Parameter):

forms_container_prosearch ()	Container für die Anzeige der erweiterten Suchfunktion im öffentlichen Bereich. Aufruf in den Vorlagen: {cp:container module="[MODULNAME]" type="prosearch" [showfields="title,description"] [limit="10"] [order="number"] [dir="asc"] [page="1"] [template="vorlage.html"] [format="html"]}	$form
forms_container_prosearch_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
forms_container_prosearch_info ()	Liefert den Titel des Containers zurück.	

6.9.77. Datei "/modules/forms/container/search.php"

Funktionen in der Datei "/modules/forms/container/search.php" (Name der Funktion, Beschreibung, Parameter):

forms_container_search ()	Container für die Anzeige der Suchfunktion im öffentlichen Bereich. Aufruf in den Vorlagen: {cp:container module="[MODULNAME]" type="search" [limit="10"] [order="number"] [dir="asc"] [template="vorlage.html"] [format="html"]}	$form
forms_container_search_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
forms_container_search_info	Liefert den Titel des Containers zurück.	

()

6.9.78. Datei "/modules/forms/datatypes/author.php"

Funktionen in der Datei "/modules/forms/datatypes/author.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.79. Datei "/modules/forms/datatypes/birthday.php"

Funktionen in der Datei "/modules/forms/datatypes/birthday.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.80. Datei "/modules/forms/datatypes/boolean.php"

Funktionen in der Datei "/modules/forms/datatypes/boolean.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.81. Datei "/modules/forms/datatypes/captcha.php"

Funktionen in der Datei "/modules/forms/datatypes/captcha.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Überprüft, ob der eingegebene Code korrekt ist. Wenn dies nicht der Fall ist, wird der Eintrag abgelehnt.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
output_random_position ()	Legt zufällige Positionsangaben in einem Array fest.	$position_array, $type
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.82. Datei "/modules/forms/datatypes/country.php"

Funktionen in der Datei "/modules/forms/datatypes/country.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.83. Datei "/modules/forms/datatypes/date.php"

Funktionen in der Datei "/modules/forms/datatypes/date.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.84. Datei "/modules/forms/datatypes/date_time.php"

Funktionen in der Datei "/modules/forms/datatypes/date_time.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.85. Datei "/modules/forms/datatypes/email.php"

Funktionen in der Datei "/modules/forms/datatypes/email.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.86. Datei "/modules/forms/datatypes/eshop.php"

Funktionen in der Datei "/modules/forms/datatypes/eshop.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form

prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.87. Datei "/modules/forms/datatypes/file.php"

Funktionen in der Datei "/modules/forms/datatypes/file.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.88. Datei "/modules/forms/datatypes/gender.php"

Funktionen in der Datei "/modules/forms/datatypes/gender.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.89. Datei "/modules/forms/datatypes/information.php"

Funktionen in der Datei "/modules/forms/datatypes/information.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.90. Datei "/modules/forms/datatypes/integer.php"

Funktionen in der Datei "/modules/forms/datatypes/integer.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.91. Datei "/modules/forms/datatypes/languages.php"

Funktionen in der Datei "/modules/forms/datatypes/languages.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.92. Datei "/modules/forms/datatypes/math.php"

Funktionen in der Datei "/modules/forms/datatypes/math.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Überprüft, ob die eingegebene Zahl korrekt ist. Wenn dies nicht der Fall ist, wird der Eintrag abgelehnt.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.93. Datei "/modules/forms/datatypes/password.php"

Funktionen in der Datei "/modules/forms/datatypes/password.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.94. Datei "/modules/forms/datatypes/password_encoded.php"

Funktionen in der Datei "/modules/forms/datatypes/password_encoded.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.95. Datei "/modules/forms/datatypes/price.php"

Funktionen in der Datei "/modules/forms/datatypes/price.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form

show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.96. Datei "/modules/forms/datatypes/relation.php"

Funktionen in der Datei "/modules/forms/datatypes/relation.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.97. Datei "/modules/forms/datatypes/section.php"

Funktionen in der Datei "/modules/forms/datatypes/section.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
create_list()	Erstellt das Menü mit der Struktur der Kategorien.	$level, $form, $count = "0"
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.98. Datei "/modules/forms/datatypes/select.php"

Funktionen in der Datei "/modules/forms/datatypes/select.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.99. Datei "/modules/forms/datatypes/select_checkboxes.php"

Funktionen in der Datei "/modules/forms/datatypes/select_checkboxes.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.100. Datei "/modules/forms/datatypes/select_multiple.php"

Funktionen in der Datei "/modules/forms/datatypes/select_multiple.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.101. Datei "/modules/forms/datatypes/select_radioboxes.php"

Funktionen in der Datei "/modules/forms/datatypes/select_radioboxes.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.102. Datei "/modules/forms/datatypes/spacer.php"

Funktionen in der Datei "/modules/forms/datatypes/spacer.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.103. Datei "/modules/forms/datatypes/states_at.php"

Funktionen in der Datei "/modules/forms/datatypes/states_at.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.104. Datei "/modules/forms/datatypes/states_ch.php"

Funktionen in der Datei "/modules/forms/datatypes/states_ch.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form

prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.105. Datei "/modules/forms/datatypes/states_de.php"

Funktionen in der Datei "/modules/forms/datatypes/states_de.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.106. Datei "/modules/forms/datatypes/states_es.php"

Funktionen in der Datei "/modules/forms/datatypes/states_es.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.107. Datei "/modules/forms/datatypes/states_fr.php"

Funktionen in der Datei "/modules/forms/datatypes/states_fr.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.108. Datei "/modules/forms/datatypes/states_it.php"

Funktionen in der Datei "/modules/forms/datatypes/states_it.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.109. Datei "/modules/forms/datatypes/text.php"

Funktionen in der Datei "/modules/forms/datatypes/text.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.110. Datei "/modules/forms/datatypes/textarea.php"

Funktionen in der Datei "/modules/forms/datatypes/textarea.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.111. Datei "/modules/forms/datatypes/time.php"

Funktionen in der Datei "/modules/forms/datatypes/time.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.112. Datei "/modules/forms/datatypes/url.php"

Funktionen in der Datei "/modules/forms/datatypes/url.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.113. Datei "/modules/forms/datatypes/wysiwyg.php"

Funktionen in der Datei "/modules/forms/datatypes/wysiwyg.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""

output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.114. Datei "/modules/forms/datatypes/wysiwyg_ckeditor.php"

Funktionen in der Datei "/modules/forms/datatypes/wysiwyg_ckeditor.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.115. Datei "/modules/forms/datatypes/wysiwyg_fck.php"

Funktionen in der Datei "/modules/forms/datatypes/wysiwyg_fck.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.116. Datei "/modules/forms/datatypes/wysiwyg_tinymce.php"

Funktionen in der Datei "/modules/forms/datatypes/wysiwyg_tinymce.php" (Name der Funktion, Beschreibung, Parameter):

check_javascript()	Clientseitige Überprüfung des Inhaltes des Datentyps.	$form
check_php()	Serverseitige Überprüfung des Inhaltes des Datentyps.	$form
clean()	Säubert den Inhalt des Datentyps vor dem Einfügen in die Datenbank.	$form
create()	Zeigt das Formular-Element für diesen Datentyp an.	$form
generate()	Formular-Felder zur Erzeugung eines neuen Feldes mit diesem Datentyp.	$form = array()
options()	Grundlegende Einstellungen zu diesem Datentyp.	$type = ""
output()	Zeigt einen Feldtypen direkt über "index.php?file=..." an, wenn dies erforderlich ist.	$form
prosearch()	Zeigt das Formular-Element für die erweiterte Suche an.	$form
search()	SQL-Befehl zur Durchsuchung eines Feldes des Datentyps.	$form
show()	Gibt den Inhalt des Feldes nach Bearbeitung aus.	$form

6.9.117. Datei "/modules/forms/filetypes/csv.php"

Funktionen in der Datei "/modules/forms/filetypes/csv.php" (Name der Funktion, Beschreibung, Parameter):

export_datasets()	Bereitet das Array für den Export der Daten vor.	$form, $data
import_datasets()	Bereitet das Array für den Import der Daten vor.	$form
options()	Grundlegende Einstellungen zu diesem Dateityp.	$type = ""
show_sample()	Liefert einen Beispieldatensatz als Array zurück.	$form

6.9.118. Datei "/modules/forms/filetypes/xml.php"

Funktionen in der Datei "/modules/forms/filetypes/xml.php" (Name der Funktion, Beschreibung, Parameter):

export_datasets()	Bereitet das Array für den Export der Daten vor.	$form, $data
import_datasets()	Bereitet das Array für den Import der Daten vor.	$form
options()	Grundlegende Einstellungen zu diesem Dateityp.	$type = ""

show_sample()	Liefert einen Beispieldatensatz als Array zurück.	$form

6.9.119. Datei "/modules/forms/templates/class.index.php"

Funktionen in der Datei "/modules/forms/templates/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_{cp:helper:folder}()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_main()	Hauptmethode, die die jeweils gewünschte Methode aufruft.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.120. Datei "/modules/forms/templates/class.webservices.php"

Funktionen in der Datei "/modules/forms/templates/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.121. Datei "/modules/forms/templates/container.php"

Funktionen in der Datei "/modules/forms/templates/container.php" (Name der Funktion, Beschreibung, Parameter):

{cp:helper:folder}_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Diese sind im Unterordner "container" jeweils als einzelne Datei abzulegen.	$form

6.9.122. Datei "/modules/forum/class.admin.php"

Funktionen in der Datei "/modules/forum/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_forum()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_category()	Zeigt das Formular zum Hinzufügen einer neuen Kategorie an.	$form
do_add_category_2()	Fügt eine neue Kategorie zu einem Forum in CP::Community hinzu.	$form
do_add_section()	Zeigt das Formular zum Hinzufügen eines neuen Forums an.	$form
do_add_section_2()	Fügt ein neues Forum zur Datenbank hinzu.	$form
do_change_position()	Ändert die Reihenfolge der Kategorien in der Datenbank.	$form
do_delete_category()	Löscht eine vorhandene Kategorie inkl. aller Themen und Beiträge.	$form
do_delete_section()	Löscht ein vorhandenes Forum inkl. aller Kategorien, Themen und Beiträge.	$form
do_edit_category()	Zeigt das Formular zum Bearbeiten einer vorhandenen Kategorie an.	$form
do_edit_category_2()	Speichert die angepassten Angaben zu einer vorhandenen Kategorie in der Datenbank.	$form
do_edit_section()	Zeigt das Formular zum Bearbeiten eines vorhandenen Forums an.	$form
do_edit_section_2()	Speichert die angepassten Angaben zu einem vorhandenen Forum in der Datenbank.	$form
do_list_sections()	Zeigt die Übersicht der Foren innerhalb der Datenbank an.	$form
do_main()	Startseite des Administrationsbereiches von CP::Community.	
do_options()	Zeigt das Formular für die Anpassung der Einstellungen an.	$form
do_options_save()	Aktualisiert die allgemeinen Einstellungen in der Datenbank.	$form
do_statistic()	Zeigt die Statistik der Foren und Mitglieder an.	$form
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.123. Datei "/modules/forum/class.base.php"

Funktionen in der Datei "/modules/forum/class.base.php" (Name der Funktion, Beschreibung, Parameter):

calculate_filesize()	Formatiert die Größeangaben von KB in das jeweils passende Format.	$filesize
check_permission()	Überprüft ob ein Benutzer eine bestimmte Aktion durchführen darf.	$form
format_actionscript()	Formatiert die in den Beiträgen enthaltenen Actionscript-Befehle.	$string
format_bbcode()	Formatiert den in den Beiträgen enthaltenen BB-Code.	$string, $mode
format_edited()	Formatiert die in den Beiträgen enthaltenen Informationen zur vorherigen Bearbeitung.	$string, $forum_user
format_include_image()	Formatiert die in den Beiträgen enthaltenen Bilder.	$string, $image_alt = ""
format_php()	Formatiert die in den Beiträgen enthaltenen PHP-Befehle.	$string
format_quote()	Formatiert die in den Beiträgen enthaltenen Zitate.	$string
format_sql()	Formatiert die in den Beiträgen enthaltenen SQL-Befehle.	$string
generate_file_path()	Erzeugt zentral die Pfade zum Forum, zu Beiträgen etc. innerhalb der PHP-Dateien.	$type, $array = array()
generate_template_path()	Gibt in Abhängigkeit von den Einstellungen den Pfad zurück, unter dem die Vorlagen liegen.	
get_images_icons()	Liefert die verfügbaren Icons für die Beiträge zurück.	$selected = ""

get_images_smilies()	Liefert die verfügbaren Smilies für die Beiträge zurück.	
get_user_posts()	Liefert die Anzahl der Beiträge eines Benutzers zurück.	$number
limit_string()	Kürzt eine URL auf die maximal erlaubte Länge.	$url_string, $url_title
secure_input()	Überprüft die Benutzereingaben und entfernt möglicherweise schadhaften Quellcode.	$string
show_posts()	Zeigt die Beiträge innerhalb eines Themas an.	$form
show_topics()	Zeigt die Themen innerhalb einer Kategorie an.	$form
validate_user()	Überprüft, ob es sich um einen eingeloggten Benutzer handelt.	

6.9.124. Datei "/modules/forum/class.index.php"

Funktionen in der Datei "/modules/forum/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_forum ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_post()	Zeigt das Formular für das Hinzufügen neuer Beiträge zu einem Thema an.	$form
do_add_post_2()	Fügt einen neuen Beitrag eines Benutzers zu einem Thema hinzu.	$form
do_add_post_status()	Zeigt eine Statusmeldung nach dem Hinzufügen eines Beitrages an.	$form
do_add_topic()	Zeigt das Formular für das Hinzufügen neuer Themen an.	$form
do_attachment()	Gibt einen Anhang für den Download an den Browser aus.	$form
do_delete_post()	Löscht einen bestehenden Beitrag eines Benutzers samt Anhang.	$form
do_delete_topic()	Löscht ein bestehendes Thema eines Benutzers samt Beiträgen und Anhängen.	$form
do_edit_post()	Zeigt das Formular für das Bearbeiten von Beiträgen zu einem Thema an.	$form
do_edit_post_2()	Aktualisiert einen bestehenden Beitrag eines Benutzers zu einem Thema.	$form
do_edit_post_status()	Zeigt eine Statusmeldung nach der Bearbeitung eines Beitrages an.	$form
do_main()	Hauptmethode, die die jeweils gewünschte Methode aufruft.	
do_search()	Zeigt die Suchfunktion für das Forum an.	$form
do_search_now()	Leitet die Suchanfrage an die Suchfunktion weiter.	$form
do_showcategory()	Zeigt eine Kategorie des Forums mit den enthaltenen Themen an.	$form
do_showmain()	Zeigt die Startseite des Forums oder ein einzelnes Forum an.	$form
do_showmaintenance()	Zeigt den Wartungsmodus an, sofern dieser aktiviert ist.	
do_shownavigation()	Zeigt die Breadcrumb-Navigation des Forums an.	$form
do_showtopic()	Zeigt ein Thema einer Kategorie mit den enthaltenen Beiträgen an.	$form
do_statistic()	Zeigt die Statistik für das Forum an.	$form
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.125. Datei "/modules/forum/class.webservices.php"

Funktionen in der Datei "/modules/forum/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.126. Datei "/modules/forum/container.php"

Funktionen in der Datei "/modules/forum/container.php" (Name der Funktion, Beschreibung, Parameter):

forum_container()	Zentrale Zusammenstellung der Container für die Ausgabe auf der Website.	$form

6.9.127. Datei "/modules/forum/container/list_categories.php"

Funktionen in der Datei "/modules/forum/container/list_categories.php" (Name der Funktion, Beschreibung, Parameter):

forum_container_list_categories ()	Container für die Anzeige eines Auswahlmenüs für die einzelnen Foren und Kategorien. Aufruf in den Vorlagen: {cp:container module="forum" type="list_categories"}	$forn
forum_container_list_categories_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.128. Datei "/modules/forum/container/list_topics.php"

Funktionen in der Datei "/modules/forum/container/list_topics.php" (Name der Funktion, Beschreibung, Parameter):

forum_container_list_topics ()	Container für die Anzeige der Themen einer Kategorie bzw. allgemein. Aufruf in den Vorlagen: {cp:container module="forum" type="list_topics" [start="0"] [limit="10"] [dir="asc"] [order="number"] [category="123"] [template="alternative.html"]}	$forn
forum_container_list_topics_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.129. Datei "/modules/forum/container/statistic.php"

Funktionen in der Datei "/modules/forum/container/statistic.php" (Name der Funktion, Beschreibung, Parameter):

forum_container_statistic ()	Container für die Anzeige verschiedener statistischer Daten zu den einzelnen Foren und Kategorien. Aufruf in den Vorlagen: {cp:container module="forum" type="statistic" [template="alternative.html"]}	$form
forum_container_statistic_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.130. Datei "/modules/friend/class.admin.php"

Funktionen in der Datei "/modules/friend/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_friend ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.131. Datei "/modules/friend/class.index.php"

Funktionen in der Datei "/modules/friend/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_friend ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_main()	Umleitung auf die Startseite, da kein Inhalt vorhanden ist.	
do_send()	Eine Email mit den Daten des Formulares wird zusammengestellt und verschickt.	$form
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.132. Datei "/modules/friend/class.webservices.php"

Funktionen in der Datei "/modules/friend/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.133. Datei "/modules/friend/container.php"

Funktionen in der Datei "/modules/friend/container.php" (Name der Funktion, Beschreibung, Parameter):

friend_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Für das "friend"-Modul ist dies lediglich die Darstellung des Formulares. Aufruf in den Vorlagen: {cp:container module="friend" [template="vorlage.html"] [return="zielseite.php"]}	$form

6.9.134. Datei "/modules/gallery/class.admin.php"

Funktionen in der Datei "/modules/gallery/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_gallery ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_category()	Zeigt das Formular zum Hinzufügen einer neuen Kategorie an.	
do_add_category_2()	Fügt eine neue Kategorie zur Datenbank hinzu.	$form
do_add_picture()	Zeigt das Formular zum Hinzufügen eines neuen Bildes zur Datenbank an.	
do_add_picture_2()	Fügt ein neues Bild zur Datenbank hinzu.	$form
do_category()	Zeigt das Formular mit der Kategorieübersicht an.	
do_choose_gallery()	Ermöglicht eine gleichzeitige Änderung mehrerer Bilder (z.B. Aktivierung, Deaktivierung).	$action, $checkbox
do_delete_category()	Löscht eine vorhandene Kategorie aus der Datenbank.	$form
do_delete_comment()	Löscht einen vorhandenen Kommentar aus der Datenbank.	$number
do_delete_picture()	Löscht ein vorhandenes Bild aus der Datenbank.	$number, $redirect = "1"
do_delete_preview_picture ()	Löscht das Vorschau-Bild aus der Datenbank und entfernt auch die entsprechenden Dateien.	$picture
do_edit_category()	Zeigt das Formular für die Bearbeitung einer vorhandenen Kategorie an.	$form
do_edit_category_2()	Aktualisiert die Angaben zu einer vorhandenen Kategorie in der Datenbank.	$form
do_edit_comment()	Zeigt das Formular zum Bearbeiten eines vorhandenen Kommentares an.	$number
do_edit_comment_2()	Aktualisiert die Angaben zu einem vorhandenen Kommentar in der Datenbank.	$form
do_edit_picture()	Zeigt das Formular für die Bearbeitung eines vorhandenen Bildes an.	$number
do_edit_picture_2()	Aktualisiert die Angaben zu einem Bild in der Datenbank.	$form
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$list, $search, $poster, $category, $search_str $page
do_options()	Zeigt das Formular für die Bearbeitung der Modul-Einstellungen an.	
do_set_options()	Aktualisiert die Einstellungen zu diesem Modul in der Datenbank.	$form

install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.135. Datei "/modules/gallery/class.base.php"

Funktionen in der Datei "/modules/gallery/class.base.php" (Name der Funktion, Beschreibung, Parameter):

do_show()	Zeigt ein einzelnes Bild über eine Vorlage an.	$row, $template = ""

6.9.136. Datei "/modules/gallery/class.index.php"

Funktionen in der Datei "/modules/gallery/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_gallery ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_comment()	Fügt einen Kommentar / eine Bewertung zu einem Bild hinzu.	$form
do_delpic()	Entfernt ein bereits vorhandenes Bild aus der Datenbank.	$number
do_delprepic()	Entfernt ein bereits vorhandenes Vorschau-Bild aus der Datenbank.	$imagefile
do_detail()	Zeigt die Detailansicht eines einzelnen Bildes an.	$number
do_edit()	Zeigt das Formular zur Bearbeitung eines vorhandenen Bildes an.	
do_editpic()	Aktualisiert die Angaben zu einem Bild in der Datenbank.	$form
do_error()	Fügt eine Information über ein fehlerhaftes Bild zur Datenbank hinzu.	$number
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$page, $dir, $category, $poster, $order
do_upload()	Zeigt das Formular zum Hinzufügen eines neuen Bildes durch einen eingeloggten Benutzer an.	
do_uploadset()	Fügt ein neues Bild zur Datenbank hinzu.	$form
frontend_editing()	Behandelt die Einblendung der Frontend-Editing-Leiste für berechtigte Benutzer.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.137. Datei "/modules/gallery/class.webservices.php"

Funktionen in der Datei "/modules/gallery/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

do_rss()	Liefert die letzten Einträge als RSS 2.0 Media-Feed aus. Aufruf: http://www.domain.de/pfad/webservices.php?file=gallery&type=rss&(limit|start|section)=...	$form
process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.138. Datei "/modules/gallery/container.php"

Funktionen in der Datei "/modules/gallery/container.php" (Name der Funktion, Beschreibung, Parameter):

gallery_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="gallery" [...]}	$forn

6.9.139. Datei "/modules/guestbook/class.admin.php"

Funktionen in der Datei "/modules/guestbook/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_guestbook ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_choose_entries()	Ermöglicht eine gleichzeitige Änderung mehrerer Einträge (z.B. Aktivierung, Deaktivierung).	$action, $checkbox
do_delete_entry()	Löscht einen Gästebuch-Eintrag aus der Datenbank.	$number
do_edit_entry()	Zeigt das Formular für die Bearbeitung eines vorhandenen Eintrages an.	$number
do_edit_entry_2()	Aktualisiert die Angaben zu einem Eintrag in der Datenbank.	$form
do_edit_entry_now()	Leitet auf das Formular für die Bearbeitung der Einträge weiter.	$number
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$form
do_options()	Zeigt das Formular für die Bearbeitung der Einstellungen des Modules an.	
do_set_options()	Aktualisiert die Einstellungen in der Datenbank.	$form
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.140. Datei "/modules/guestbook/class.index.php"

Funktionen in der Datei "/modules/guestbook/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_guestbook	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	

()		
do_add_entry()	Fügt einen neuen Eintrag zur Datenbank hinzu.	$form
do_getsmilies()	Generiert eine Übersicht mit der Auswahl der verfügbaren Smilies / Emoticons.	$sPath
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$form
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.141. Datei "/modules/guestbook/class.webservices.php"

Funktionen in der Datei "/modules/guestbook/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.142. Datei "/modules/guestbook/container.php"

Funktionen in der Datei "/modules/guestbook/container.php" (Name der Funktion, Beschreibung, Parameter):

guestbook_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="guestbook" [...]}	$forn

6.9.143. Datei "/modules/help/class.admin.php"

Funktionen in der Datei "/modules/help/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_help()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_entry()	Formular für das Hinzufügen eines neuen Eintrags.	$section = ""
do_add_entry_2()	Fügt den neuen Eintrag in die Datenbank ein.	$form
do_add_section()	Formular zum Hinzufügen einer neuen Kategorie.	
do_add_section_2()	Fügt eine neue Kategorie in die Datenbank ein.	$form
do_delete_entry()	Löscht einen Eintrag aus der Datenbank.	$number
do_delete_section()	Löscht eine vorhandene Kategorie samt den enthaltenen Einträgen.	$number
do_details()	Darstellung eines einzelnen Eintrages.	$number
do_edit_entry()	Formular für das Bearbeiten eines vorhandenen Eintrages.	$number
do_edit_entry_2()	Aktualisiert einen vorhandenen Eintrag in der Datenbank.	$form
do_edit_section()	Formular für die Bearbeitung einer vorhandenen Kategorie.	$number
do_edit_section_2()	Aktualisiert eine vorhandene Kategorie in der Datenbank.	$form
do_get_section()	Liefert eine Angabe zu einer bestimmten Kategorie zurück.	$number, $field = "title"
do_main()	Generiert die Hauptansicht des Modules.	
do_move()	Erzeugt den Fussbereich für dieses Modul.	$type, $number
do_print()	Zeigt den Button zum Druck der Anleitung an.	
do_search()	Führt eine Suche in der Anleitung nach passenden Einträgen durch.	$search, $select_box, $select_section, $page
do_search_now()	Leitet auf die eigentliche Suche weiter.	$search, $select_box, $select_section, $page
do_searchbox()	Zeigt das Suchformular zur Suche in einer Kategorie bzw. insgesamt an.	$search, $select_box, $select_section
do_section()	Darstellung der Einträge in einer Kategorie.	$number, $page = "1"
do_shortnavigation()	Zeigt die Kurznavigation auf der linken Seite an.	
do_show()	Wechselt zwischen der Anzeige für die Redakteure und der kompletten Anzeige.	$show
do_update_help()	Gleicht die Anleitung mit der aktuell online verfügbaren Version ab.	
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.144. Datei "/modules/help/class.base.php"

Funktionen in der Datei "/modules/help/class.base.php" (Name der Funktion, Beschreibung, Parameter):

process()	contentpapst_help::process Verarbeitet die Einträge der Anleitung und gibt diese zurück.	$content
process_html()	contentpapst_help::process_html Verarbeitet einen Abschnitt und gibt diesen als HTML-Quellcode aus.	$content
process_include()	contentpapst_help::process_include Bindet eine externe Datei ein und liefert deren Inhalt zurück.	$filename, $type = "txt"
process_notice()	contentpapst_help::process_notice Erstellt die Hinweisfenster innerhalb der Einträge.	$content
process_table()	contentpapst_help::process_table Erstellt die Tabellen innerhalb der Einträge.	$title, $content
process_url()	contentpapst_help::process_url Erstellt die URL's innerhalb der Einträge.	$url, $content

6.9.145. Datei "/modules/help/class.index.php"

Funktionen in der Datei "/modules/help/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_help()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
metatags()	Behandelt die Einblendung der Metatags.	

search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.146. Datei "/modules/help/class.webservices.php"

Funktionen in der Datei "/modules/help/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.147. Datei "/modules/help/container.php"

Funktionen in der Datei "/modules/help/container.php" (Name der Funktion, Beschreibung, Parameter):

help_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="help" [...]}	$forn

6.9.148. Datei "/modules/languages/class.admin.php"

Funktionen in der Datei "/modules/languages/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_languages ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_language()	Fügt die Sprachdateien für eine neue Sprache hinzu.	$form, $upload
do_delete_language()	Löscht eine Sprache komplett aus dem System.	$form
do_edit_language()	Bearbeitung einer vorhandenen Sprache.	$form
do_edit_language_file()	Bearbeitung einer einzelnen Sprachdatei einer vorhandenen Sprache.	$form
do_edit_language_file_2 ()	Speichert die geänderte Sprachdatei einer einzelnen Sprache ab.	$form
do_main()	Generiert die Hauptansicht des Modules.	
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.149. Datei "/modules/languages/class.base.php"

Funktionen in der Datei "/modules/languages/class.base.php" (Name der Funktion, Beschreibung, Parameter):

get_language_files()	Ermittelt die im System pro Sprache vorhandenen Sprachdateien.	
get_languages()	Ermittelt die im System vorhandenen Sprachen.	
get_possible_languages ()	Liefert die nach ISO 639-1 möglichen Sprachen.	

6.9.150. Datei "/modules/languages/class.index.php"

Funktionen in der Datei "/modules/languages/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_languages ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.151. Datei "/modules/languages/class.webservices.php"

Funktionen in der Datei "/modules/languages/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.152. Datei "/modules/languages/container.php"

Funktionen in der Datei "/modules/languages/container.php" (Name der Funktion, Beschreibung, Parameter):

languages_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="languages" [...]}	$forn

6.9.153. Datei "/modules/links/class.admin.php"

Funktionen in der Datei "/modules/links/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_links()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_category()	Zeigt das Formular zum Hinzufügen einer neuen Kategorie an.	
do_add_category_2()	Fügt eine neue Kategorie zur Datenbank hinzu.	$form

do_add_link()	Zeigt das Formular zum Hinzufügen eines neuen Links an.	
do_add_link_2()	Fügt einen neuen Link zur Datenbank hinzu.	$form
do_categories()	Zeigt eine Übersicht der aktuell vorhandenen Kategorien an.	
do_check()	Überprüft, ob die in der Datenbank eingetragenen Links noch verfügbar sind.	$form
do_choose_entries()	Ermöglicht eine gleichzeitige Änderung mehrerer Einträge (z.B. Aktivierung, Deaktivierung).	$action, $checkbox
do_delete_category()	Löscht einen vorhandenen Kommentar aus der Datenbank.	$number
do_delete_comment()	Löscht einen vorhandenen Kommentar aus der Datenbank.	$number
do_delete_link()	Löscht einen vorhandenen Link aus der Datenbank.	$number
do_edit_category()	Zeigt das Formular für die Bearbeitung einer vorhandenen Kategorie an.	$number
do_edit_category_2()	Aktualisiert die Angaben zu einer Kategorie in der Datenbank.	$form
do_edit_comment()	Zeigt das Formular zum Bearbeiten eines vorhandenen Kommentares an.	$number
do_edit_comment_2()	Aktualisiert die Angaben zu einem Kommentar in der Datenbank.	$form
do_edit_link()	Zeigt das Formular zum Bearbeiten eines vorhandenen Links an.	$number
do_edit_link_2()	Aktualisiert die Angaben zu einem Link in der Datenbank.	$form
do_filelist()	Liefert eine Auflistung der aktuell verfügbaren Vorlagen für die Links zurück.	$sPath, $sFile
do_main()	Generiert die Hauptansicht des Modules Links.	$form
do_options()	Zeigt das Formular für die Bearbeitung der Einstellungen des Modules an.	
do_search_now()	Leitet vom Suchformular auf die eigentliche Ergebnisseite weiter.	$form
do_set_options()	Aktualisiert die Angaben zu einem Modul in der Datenbank.	$form
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.154. Datei "/modules/links/class.base.php"

Funktionen in der Datei "/modules/links/class.base.php" (Name der Funktion, Beschreibung, Parameter):

do_menu()	contentpapst_links::do_menu Über diese Funktion wird ein Array der Link-Kategorien samt Unterkategorien zurückgeliefert.	$level, $count = "0"
do_show()	contentpapst_links::do_show Darstellung eines einzelnen Eintrages im Link-Modul.	$number, $mode

6.9.155. Datei "/modules/links/class.index.php"

Funktionen in der Datei "/modules/links/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_links()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_comment()	Fügt einen neuen Kommentar zu einem Link in die Datenbank ein.	$form
do_add_link()	Zeigt das Formular zum Hinzufügen eines neuen Links an.	
do_add_link_2()	Fügt einen neuen Link in die Datenbank ein.	$form
do_category()	Zeigt die Auflistung der Kategorien in einer Übersicht an.	$masscat, $step
do_details()	Zeigt einen Link im Detail an (u.a. mit einem Formular für die Bewertung).	$number
do_error_link()	Markiert einen Link in der Datenbank als fehlerhaft (z.B. wenn die URL nicht mehr stimmt).	$number
do_fav_links()	Zeigt die Auflistung der am besten bewerteten Links an.	
do_header()	Zeigt die obere Leiste mit den Links zu den einzelnen Bereichen an.	
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	
do_navigation()	Generiert die Breadcrumb-Navigation für den Seitenkopf.	$nav_array, $no_break = ""
do_new_links()	Zeigt die Auflistung der erst kürzlich hinzugefügten Links an.	
do_out()	Zeichnet einen Klick auf einen Link auf und leitet anschließend auf die externe Website weiter.	$number, $url
do_search()	Zeigt die Suchfunktion für die Links an.	
do_section()	Zeigt eine Kategorie mit den darin enthaltenen Links an.	$number, $page
do_show_category()	Zeigt eine einzelne Kategorie über eine Vorlage an.	$sec, $masscat, $step
do_top_links()	Zeigt die Auflistung der am häufigsten aufgerufenen Links an.	
do_vote()	Zeigt das Formular für die Bewertung eines Links an.	$number
do_voteset()	Fügt eine neue Bewertung zu einem Link in die Datenbank ein.	$form
frontend_editing()	Behandelt die Einblendung der Frontend-Editing-Leiste für berechtigte Benutzer.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.156. Datei "/modules/links/class.webservices.php"

Funktionen in der Datei "/modules/links/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.157. Datei "/modules/links/container.php"

Funktionen in der Datei "/modules/links/container.php" (Name der Funktion, Beschreibung, Parameter):

links_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Dies sind u.a. die verschiedenen vorgefertigen Navigationstypen. Aufruf in den Vorlagen: {cp:container module="navigation" [...]}	$form
links_details()	Hilfsfunktion für die Darstellung von Links über die Container.	$number, $mode, $template = ""

6.9.158. Datei "/modules/media/class.admin.php"

Funktionen in der Datei "/modules/media/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_media ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_detail()	Generiert die Detailansicht für eine einzelne Datei.	$object
do_file_error()	Gibt eine Fehlermeldung bei Problemen mit dem Upload aus.	
do_getdirs()	Ermittelt die vorhandenen Verzeichnisse in einem bestimmten Bereich.	$sPath
do_getfiles()	Ermittelt die vorhandenen Dateien in einem bestimmten Bereich.	$sPath
do_graphic()	Generiert die verkleinerten Vorschaubilder der Dateien "on-the-fly".	$file_name, $height, $width, $file_extension
do_main()	Hauptansicht für das Modul Medienbibliothek mit einzelnen Formularen.	$path
do_search()	Zentrale Suchfunktion für die einzelnen Unterordner des Dateisystems.	$form
do_search_now()	Leitet vom Suchformular auf die eigentliche Suche weiter.	$form
do_set_rights()	Setzt die neuen Zugriffsrechte (CHMOD) für eine Datei (momentan nur auf Linux/Unix-Systemen).	$form
do_shortnavigation()	Zeigt die Kurznavigation auf der linken Seite an.	
do_upload()	Führt den Upload der Dateien in das aktuelle Verzeichnis durch.	$path
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.159. Datei "/modules/media/class.index.php"

Funktionen in der Datei "/modules/media/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_media ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.160. Datei "/modules/media/class.webservices.php"

Funktionen in der Datei "/modules/media/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.161. Datei "/modules/media/container.php"

Funktionen in der Datei "/modules/media/container.php" (Name der Funktion, Beschreibung, Parameter):

media_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="media" [...]}	$forn

6.9.162. Datei "/modules/navigation/class.admin.php"

Funktionen in der Datei "/modules/navigation/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_navigation ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_article_structure ()	Legt neue Navigationspunkte auf der Basis der Artikelstruktur an.	$form
do_add_article_structure_now ()	Legt neue Navigationspunkte auf der Basis der Artikelstruktur an.	$number = "0", $target = "0", $position_count = "999"
do_add_menu()	Zeigt das Menü für die Auswahl eines Zieles für einen Navigationspunkt an.	
do_add_navigation()	Fügt einen neuen Navigationspunkt in die Datenbank ein.	$form
do_delete_navigation()	Löscht einen Navigationspunkt samt dessen Unterpunkten.	$form
do_edit_navigation()	Formular für die Bearbeitung eines vorhandenen Navigationspunktes.	$form
do_edit_navigation_2()	Aktualisiert einen Navigationspunkt in der Datenbank.	$form
do_edit_navigation_now ()	Leitet auf das Formular für die Bearbeitung eines Navigationspunktes weiter.	$form
do_javascript()	Liefert eine Javascript-Funktion zum Öffnen eines Fensters.	$form_name = "window_navigation_addmenu"
do_main()	Hauptansicht für das Modul Navigation mit einzelnen Formularen.	$form
do_menu()	Erstellt das Navigationsmenü der in der Datenbank vorhandenen	$level, $count = "0"

	Navigationspunkte.	
do_menu_options()	Erstellt die Auswahl der vorhandenen Navigationspunkte.	$level, $count = "0", $active = "0", $base = "0"
do_move()	Ändert die Positionen der vorhandenen Navigationspunkte.	$form
do_show()	Zeigt den <iframe> zusammen mit den Navigationsmenüs an.	
do_sort()	Ordnet die Unterpunkte eines Navigationspunktes z.B. aufsteigend alphabetisch.	$form
do_update()	Aktualisiert die Reihenfolge der Navigationspunkte.	$navigation_list
do_update_now()	Aktualisiert die Reihenfolge der Navigationspunkte.	$form
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.163. Datei "/modules/navigation/class.index.php"

Funktionen in der Datei "/modules/navigation/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_navigation ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.
metatags()	Behandelt die Einblendung der Metatags.
search()	Stellt die Verbindung zur Volltextsuche her.

6.9.164. Datei "/modules/navigation/class.webservices.php"

Funktionen in der Datei "/modules/navigation/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.165. Datei "/modules/navigation/container.php"

Funktionen in der Datei "/modules/navigation/container.php" (Name der Funktion, Beschreibung, Parameter):

navigation_block_entry ()	Überprüft, ob ein Navigationspunkt nicht angezeigt werden kann, da der Benutzer / Kunde eingeloggt bzw. nicht eingeloggt ist.	$status, $admin
navigation_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Dies sind u.a. die vorgefertigten Navigationstypen oder komplett eigene Navigationen. Aufruf in den Vorlagen: {cp:container module="navigation" [...]}	$form
navigation_get_ids()	Ermittelt die ausgewählten Navigationspunkte ausgehend vom untersten Navigationspunkt.	$form, $this_id
navigation_get_root()	Ermittelt den Hauptnavigationspunkt für die Anzeige der entsprechenden Vorlage.	$form

6.9.166. Datei "/modules/news/class.admin.php"

Funktionen in der Datei "/modules/news/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_news ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_category()	Zeigt das Formular zum Hinzufügen einer neuen Kategorie an.	
do_add_category_2()	Fügt eine neue Kategorie zur Datenbank hinzu.	$form
do_add_news()	Zeigt das Formular zum Hinzufügen einer neuen News an.	$form
do_add_news_2()	Fügt eine neue News zur Datenbank hinzu.	$form
do_blocked_contact()	Zeigt ein Formular an, wenn eine News bereits durch einen anderen Benutzer bearbeitet wird.	$number
do_blocked_contact_2()	Versendet die Information an den anderen Benutzer per Email.	$form
do_category()	Zeigt die Übersicht über die Kategorien an.	
do_choose_news()	Ermöglicht eine gleichzeitige Änderung mehrerer News (z.B. Aktivierung, Deaktivierung).	$action, $checkbox
do_delete_category()	Löscht eine vorhandene Kategorie aus der Datenbank.	$number
do_delete_comment()	Löscht einen vorhandenen Kommentar zu einer News aus der Datenbank.	$number
do_delete_news()	Löscht eine vorhandene News aus der Datenbank samt der damit verknüpften Kommentare.	$number, $redirect = "1"
do_duplicate_news()	Kopiert eine bereits vorhandene News.	$number, $redirect = "1"
do_edit_category()	Zeigt das Formular für die Bearbeitung einer vorhandenen Kategorie an.	$number
do_edit_category_2()	Aktualisiert die Angaben zu einer Kategorie in der Datenbank.	$form
do_edit_comment()	Zeigt das Formular für die Bearbeitung eines vorhandenen Kommentares an.	$number
do_edit_comment_2()	Aktualisiert die Angaben zu einem Kommentar in der Datenbank.	$form
do_edit_news()	Zeigt das Formular für die Bearbeitung einer vorhandenen News an.	$number
do_edit_news_2()	Aktualisiert die Angaben zu einer vorhandenen News in der Datenbank.	$form
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$form
do_main_results()	Holt die zutreffenden News aus der Datenbank.	$form
do_options()	Zeigt das Formular für die Bearbeitung der Einstellungen des Modules an.	
do_set_options()	Aktualisiert die Einstellungen zu diesem Modul in der Datenbank.	$form
do_shortnavigation()	Zeigt die Kurznavigation auf der linken Seite an.	
install()	SQL- und Dateianweisungen zur Installation des Modules.	

ping()	Führt einen Ping zu einer festgelegten URL durch.	$host, $title, $url, $params = ""
ping_blo_gs()	Führt einen Ping zu http://www.blo.gs/ durch.	
ping_blogg_de()	Führt einen Ping zu http://www.blogg.de/ durch.	
ping_google()	Führt einen Ping zu http://www.google.com/ durch.	
ping_pingoat()	Führt einen Ping zu http://www.pingoat.com/ durch. Alternative: http://pinger.blogflux.com/	
ping_pingomatic()	Führt einen Ping zu http://www.pingomatic.com/ durch.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.167. Datei "/modules/news/class.base.php"

Funktionen in der Datei "/modules/news/class.base.php" (Name der Funktion, Beschreibung, Parameter):

do_menu()	contentpapst_news::do_menu Über diese Funktion wird ein Array der Link-Kategorien samt Unterkategorien erstellt und zurückgeliefert.	$level, $count = "0"
do_show()	contentpapst_news::do_show Über diese Funktion werden die News u.a. für Ausgabe über die Container formatiert.	$form, $no_path = "0"

6.9.168. Datei "/modules/news/class.index.php"

Funktionen in der Datei "/modules/news/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_news ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_archive()	Zeigt ein Archiv mit den bislang erschienenen News an.	$form
do_archive_now()	Leitet vom Suchformular auf das eigentliche News-Archiv weiter.	$form
do_category()	Zeigt die News in einer bestimmen Kategorie zusammen mit der Beschreibung der Kategorie an.	$number
do_comment()	Fügt einen Kommentar zu einer News in die Datenbank ein.	$form, $number
do_detail()	Zeigt eine News in einer Detailansicht an.	$number
do_friend()	Zeigt ein Formular für die Weiterempfehlung einer News an.	$number
do_friend_send()	Verschickt die Weiterempfehlung einer News per Email.	$form
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$days
do_send()	Zeigt das Formular für die Erstellung einer neuen News durch einen Benutzer an.	
do_send_2()	Fügt die News eines Benutzers in die Datenbank ein.	$form
frontend_editing()	Behandelt die Einblendung der Frontend-Editing-Leiste für berechtigte Benutzer.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.169. Datei "/modules/news/class.webservices.php"

Funktionen in der Datei "/modules/news/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

do_rss()	Liefert die letzten Einträge als RSS 2.0 aus. Aufruf: http://www.domain.de/pfad/webservices.php?file=news&type=rss&(dir|limit|order)=...	$form
process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.170. Datei "/modules/news/container.php"

Funktionen in der Datei "/modules/news/container.php" (Name der Funktion, Beschreibung, Parameter):

news_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="news" [...]}	$form

6.9.171. Datei "/modules/newsletter/class.admin.php"

Funktionen in der Datei "/modules/newsletter/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_newsletter ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_issue()	Verschickt einen Newsletter allgemein bzw. nur eine Testmail an den Administrator.	$form, $action, $articledesc = "", $articletext = ""
do_add_member()	Zeigt das Formular zur Hinzufügen eines neuen Mitgliedes an.	
do_add_member_2()	Fügt ein neues Mitglied zur Datenbank hinzu.	$form
do_add_newsletter()	Zeigt die zweite Seite zum Hinzufügen eines neuen Newsletters an.	$form
do_add_newsletter_2()	Zeigt die dritte Seite zum Hinzufügen eines neuen Newsletters an.	$form
do_add_newsletter_3()	Fügt einen neuen Newsletter in die Datenbank ein.	$form
do_choose_member()	Ermöglicht eine gleichzeitige Änderung mehrerer Mitglieder (z.B. Aktivierung, Deaktivierung).	$action, $checkbox
do_delete_issue()	Löscht eine bereits vorhandene Newsletter-Ausgabe aus der Datenbank.	$number

do_delete_member()	Löscht ein bereits vorhandenes Mitglied aus der Datenbank.	$number
do_delete_news()	Löscht einen bereits vorhandenen Newsletter aus der Datenbank.	$number
do_delete_tracking()	Löscht einen bereits vorhandenen Tracking-Code aus der Datenbank.	$number
do_edit_issue()	Zeigt das Formular für die Bearbeitung einer Newsletter-Ausgabe an.	$number
do_edit_issue_2()	Aktualisiert die Angaben zu einer Newsletter-Ausgabe in der Datenbank.	$form
do_edit_member()	Zeigt das Formular zur Bearbeitung eines vorhandenen Mitgliedes an.	$number
do_edit_member_2()	Aktualisiert die Angaben zu einem vorhandenen Mitglied in der Datenbank.	$form
do_edit_news()	Zeigt das Formular zur Bearbeitung eines vorhandenen Newsletters an.	$number
do_edit_news_2()	Aktualisiert die Angaben zu einem vorhandenen Newsletter in der Datenbank.	$form
do_edit_tracking()	Zeigt das Formular zur Bearbeitung eines vorhandenen Tracking-Codes an.	$number
do_edit_tracking_2()	Aktualisiert die Angaben zu einem vorhandenen Tracking-Code in der Datenbank.	$form
do_filter()	Begriffsfilter für Email-Adressen und Namen, die bei der Anmeldung automatisch abgelehnt werden.	
do_filter_2()	Fügt eine neue Filter-Regel zur Datenbank hinzu.	$form = array()
do_filter_3()	Entfernt eine vorhandene Filter-Regel aus der Datenbank.	$form = array()
do_import_member()	Import die Email-Adressen neuer Mitglieder aus einer angegebenen Datei.	$form
do_invitation()	Schickt eine Einladung an neue Mitglieder per Email raus.	$form
do_issue_test()	Zeigt die Ergebnisseite nach dem Versand einer Testmail an.	
do_javascript()	Javascript-Code für den WYSIWYG-Editor des Modules.	
do_link_tracking()	Fügt einen neuen Tracking-Code zur Datenbank hinzu.	$linkurl
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$form
do_member()	Zeigt die Mitglieder aus allen Newslettern in einer Übersicht an.	$form
do_member_results()	Zeigt die Mitglieder aus allen Newslettern in einer Übersicht an.	$form
do_newsletter()	Zeigt die Übersicht der bereits vorhandenen Newsletter sowie ein Formular zum Hinzufügen eines neuen Newsletters an.	
do_options()	Zeigt das Formular für die Bearbeitung der Modul-Einstellungen an.	
do_send_email()	Verschickt eine Newsletter-Ausgabe an die für diesen Newsletter angemeldeten Mitglieder.	$form
do_send_member()	Verschickt eine Email an ein einzelnes Mitglied.	$form
do_set_options()	Aktualisiert die Einstellungen zu diesem Modul in der Datenbank.	$form
do_tracking()	Zeigt die Übersicht der in den Newsletter-Ausgaben getrackten URL's samt der angefallenen Klicks an.	$form
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.172. Datei "/modules/newsletter/class.index.php"

Funktionen in der Datei "/modules/newsletter/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_newsletter ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_friend()	Verschickt die Weiterempfehlung eines Newsletters per Email.	$form
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$section
do_read()	Zeigt einen einzelnen Newsletter in einer Detailansicht an.	$number
do_search()	Durchsucht die eingetragenen Newsletter nach bestimmten Begriffen.	$search
do_send()	Fügt einen neuen Abonnenten zu einem Newsletter hinzu oder entfernt diesen wieder.	$action, $username, $useremail, $key, $bdate_1, $bdate_2, $bdate_3, $forename, $gender, $selected
do_url()	Zeichnet den Klick auf einen Tracking-Link auf und leitet anschließend zur Ziel-URL weiter.	$number, $hash
frontend_editing()	Behandelt die Einblendung der Frontend-Editing-Leiste für berechtigte Benutzer.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.173. Datei "/modules/newsletter/class.webservices.php"

Funktionen in der Datei "/modules/newsletter/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.174. Datei "/modules/newsletter/container.php"

Funktionen in der Datei "/modules/newsletter/container.php" (Name der Funktion, Beschreibung, Parameter):

newsletter_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="newsletter" [...]}	$forn

6.9.175. Datei "/modules/options/class.admin.php"

Funktionen in der Datei "/modules/options/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_options ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_bookmark_now ()	Speichert den neuen Bookmark in der Datenbank ab.	$form
do_add_note_now()	Speichert die neuen Notiz in der Datenbank ab.	$form
do_add_site()	Fügt einen neuen Mandanten zum System hinzu. Kopiert werden für diesen neben den Dateien auch die Einstellungen in der Datenbank und es wird eine neue Kategorie für die Systemmeldungen im Aufgaben-Modul angelegt.	$form
do_add_site_error()	Zeigt die Fehlerseite beim Versuch des Hinzufügens eines neuen Mandanten.	$form
do_add_site_success()	Zeigt die Ergebnisseite nach dem Hinzufügen eines neuen Mandanten.	
do_check_permalink()	Liefert die Information zurück, ob ein neu eingegebener Permalink genutzt werden kann oder nicht.	
do_cleanup_options()	Überprüft, ob für alle Mandanten alle Einstellungen des Standardmandanten vorhanden sind. Wenn nicht, dann werden diese hinzugefügt.	
do_delete_bookmark()	Löscht einen bereits vorhandenen Persönlichen Bookmark eines Benutzers.	$numbe
do_delete_bookmarks()	Löscht bereits vorhandene Persönlichen Bookmarks eines Benutzers.	$form
do_delete_cache()	Löscht alle im Cache abgelegten Dateien.	$form
do_delete_log()	Löscht die gesamten bisher gespeicherten Aktivitäten aus dem Log.	
do_delete_note()	Löscht eine bereits vorhandene Persönliche Notiz eines Benutzers.	$numbe
do_delete_notes()	Löscht bereits vorhandene Persönliche Notizen eines Benutzers.	$form
do_delete_site()	Löscht einen Mandanten inklusive aller Inhalte, Vorlagen und sontiger Dateien.	$form
do_edit_site()	Zeigt das Formular zur Bearbeitung eines vorhandenen Mandanten an.	$form
do_edit_site_2()	Aktualisiert die Angaben zu einem vorhandenen Mandanten in der Datenbank und in den Konfigurationsdateien.	$form
do_install()	Installiert ein neues Modul und aktualisiert die Datenbank.	$module
do_list_active()	Generiert die Seite unter "Einstellungen > Modulaktivierung".	
do_list_cache()	Generiert die Seite unter "Einstellungen > Cache".	
do_list_contact()	Generiert die Seite unter "Einstellungen > Kontakt".	
do_list_gears()	Generiert die Seite unter "Einstellungen > Verwaltung beschleunigen mit Gears".	
do_list_general()	Generiert die Seite unter "Einstellungen > Allgemeine Einstellungen".	
do_list_logging()	Generiert die Seite unter "Einstellungen > Logging".	$form
do_list_logging_now()	Wechselt nach der Einschränkung der Suche unter "Einstellungen > Logging" zurück auf das Logging.	$form
do_list_metatags()	Generiert die Seite unter "Einstellungen > Metatags".	
do_list_phpinfo()	Zeigt die Ausgabe von phpinfo() in einem Iframe an.	
do_list_search()	Generiert die Seite unter "Einstellungen > Volltextsuche".	
do_list_server()	Generiert die Seite unter "Einstellungen > Server".	
do_list_sitemaps()	Zeigt die Vorschaltseite vor der Generierung der Sitemap-Datei im Sitemap-Protokoll an.	$form
do_list_sitemaps_now()	Generiert die Datei "sitemap-[MANDANTENNAME].xml" bzw. "sitemap-[MANDANTENNAME].xml.gz" für das Sitemap-Protokoll. -> Weitere Informationen zum Sitemap-Protokoll: https://www.sitemaps.org/ -> Feld "Priorität": Priorität der Seiten zwischen 0.0 (niedrigste) und 1.0 (höchste) -> Feld "Aktualisierungszeit": "always", "hourly", "daily", "weekly", "monthly" oder "yearly"	$form
do_list_sites()	Generiert die Seite unter "Einstellungen > Mandatenverwaltung".	
do_list_updates()	Zeigt die für das System momentan verfügbaren Updates und Erweiterungen an.	
do_list_urls()	Generiert die Seite unter "Einstellungen > Suchmaschinenfreundliche URL's".	$form
do_list_urls_permalinks ()	Erzeugt die Auflistung der Permalinks eines Mandanten unter "Einstellungen > Suchmaschinenfreundliche URL's".	$form
do_main()	Generiert die Hauptseite unter "Einstellungen" mit der Übersicht verfügbaren Unterseiten.	
do_manage_bookmarks ()	Zeigt das Formular zum Hinzufügen eines persönlichen Bookmarks sowie die bisher bereits vorhandenen Bookmarks an.	$form
do_manage_notes()	Zeigt das Formular zum Hinzufügen einer persönlichen Notiz sowie die bisher bereits vorhandenen Notizen an.	$form
do_modules()	Zeigt die Übersichtsseite der Module an.	
do_service()	Zeigt die Formulare im Bereich Kundenservice an.	
do_service_email()	Versendet die Supportanfrage an den den Hersteller der Software.	$form
do_service_success()	Generiert die Ergebnisseite nach dem erfolgreichen Versand.	
do_service_text()	Liefert den Titel für eine Update-Mail.	
do_set_active()	Ändert den Status eines vorhandenen Modules.	$form
do_set_cache()	Speichert die Einstellungen von "Einstellungen > Cache" in der Datenbank ab.	$form
do_set_contact()	Speichert die Einstellungen von "Einstellungen > Kontakt" in der Datenbank ab.	$form
do_set_gears()	Speichert die Einstellungen von "Einstellungen > Verwaltung beschleunigen mit Gears" in der Datenbank ab.	$form
do_set_general()	Speichert die Einstellungen von "Einstellungen > Allgemeine Einstellungen" in der Datenbank ab.	$form
do_set_logging()	Speichert die Einstellungen von "Einstellungen > Logging" in der Datenbank ab.	$form
do_set_metatags()	Speichert die Einstellungen von "Einstellungen > Metatags" in der Datenbank ab.	$form
do_set_search()	Speichert die Einstellungen von "Einstellungen > Volltextsuche" in der Datenbank ab.	$form
do_set_site()	Wechselt den aktuellen Standard-Mandanten des Systems.	$form
do_set_urls()	Speichert die Einstellungen von "Einstellungen > Suchmaschinenfreundliche URL's" in der Datenbank ab.	$form
do_update_notification()	Blendet die Update-Benachrichtigung auf der Startseite bis zum Erscheinen der nächsten Version aus.	$versior
do_update_site()	Aktualisiert die Dateien eines Mandanten auf einem externen FTP-Server.	$form
get_modules()	Gibt eine Liste der im System vorhandenen Module zurück.	$path

install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.176. Datei "/modules/options/class.base.php"

Funktionen in der Datei "/modules/options/class.base.php" (Name der Funktion, Beschreibung, Parameter):

do_customize()	Passt das Layout / die Funktionsweise des Administrationsbereiches an Benutzerwünsche an (z.B. linke Spalte ein-/ausblenden). Aufruf: /admin.php?file=options&mode=customize&form[file]=modul&form[name]=optionsname&form[value]=wert	$form
do_mod_rewrite()	Liefert eine Liste der abhängig von den installierten Modulen benötigten RewriteRules für "mod_rewrite" zurück.	
do_permalinks()	contentpapst_options::do_permalinks Kopiert die Permalinks aus den einzelnen Datenbanktabellen in die zentrale Datenbanktabelle *_permalinks.	$table = ""
do_start()	contentpapst_options::do_start Zeigt die Startseite des Administrationsbereiches an.	

6.9.177. Datei "/modules/options/class.index.php"

Funktionen in der Datei "/modules/options/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_options ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_tag()	Fügt einen neuen Tag zur Datenbank hinzu.	$form
do_frontend_editing()	Stellt den Javascript-Code für das Frontend-Editing bereit.	
do_frontend_editing_field ()	Stellt das Formular für die Bearbeitung des Inhaltes eines Feldes per Frontend-Editing bereit.	$form
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$section
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.178. Datei "/modules/options/class.webservices.php"

Funktionen in der Datei "/modules/options/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.179. Datei "/modules/options/container.php"

Funktionen in der Datei "/modules/options/container.php" (Name der Funktion, Beschreibung, Parameter):

options_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="options" [...]}	$forn

6.9.180. Datei "/modules/options/container/facebook_activity_feed.php"

Funktionen in der Datei "/modules/options/container/facebook_activity_feed.php" (Name der Funktion, Beschreibung, Parameter):

options_container_facebook_activity_feed()	Container für die Ausgabe des Codes de Activity Feed von Facebook (http://developers.facebook.com/plugins). Aufruf in den Vorlagen: {cp:container module="options" type="facebook_activity_feed" [...]}	$forn
options_container_facebook_activity_feed_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
options_container_facebook_activity_feed_info()	Liefert den Titel des Containers zurück.	

6.9.181. Datei "/modules/options/container/facebook_like.php"

Funktionen in der Datei "/modules/options/container/facebook_like.php" (Name der Funktion, Beschreibung, Parameter):

options_container_facebook_like ()	Container für die Ausgabe des Codes des Like-Buttons von Facebook (http://developers.facebook.com/plugins). Aufruf in den Vorlagen: {cp:container module="options" type="facebook_like" [...]}	$forn
options_container_facebook_like_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
options_container_facebook_like_info ()	Liefert den Titel des Containers zurück.	

6.9.182. Datei "/modules/options/container/facebook_recommendations.php"

Funktionen in der Datei "/modules/options/container/facebook_recommendations.php" (Name der Funktion, Beschreibung, Parameter):

options_container_facebook_recommendations()	Container für die Ausgabe des Codes der Recommendations von Facebook (http://developers.facebook.com/plugins). Aufruf in den Vorlagen: {cp:container module="options" type="facebook_recommendations" [...]}	$form
options_container_facebook_recommendations_config()	Liefert Informationen über den Aufbau des Containers zurück.	
options_container_facebook_recommendations_info()	Liefert den Titel des Containers zurück.	

6.9.183. Datei "/modules/options/container/frontend_editing_fields.php"

Funktionen in der Datei "/modules/options/container/frontend_editing_fields.php" (Name der Funktion, Beschreibung, Parameter):

options_container_frontend_editing_fields()	Container für die Erzeugung der eindeutigen ID für die Bearbeitung der Inhalte einzelner Feldinhalte auf der Webseite. Aufruf in den Vorlagen: {cp:container module="options" type="frontend_editing_fields" table="[TABLE]" field="[FIELD]" field_type="[TYPE]" number="[NUMBER]"}	$form
options_container_frontend_editing_fields_config()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.184. Datei "/modules/pdf/class.admin.php"

Funktionen in der Datei "/modules/pdf/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_pdf()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_create_pdf()	Stellt die gewünschten Inhalte zusammen und startet die Generierung.	$form, $pdf_articlesection, $pdf_article, $pdf_newssection, $pdf_news
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	
do_set_options()	Aktualisiert die Einstellungen in der Datenbank.	$form
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.185. Datei "/modules/pdf/class.index.php"

Funktionen in der Datei "/modules/pdf/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_pdf()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.
metatags()	Behandelt die Einblendung der Metatags.
search()	Stellt die Verbindung zur Volltextsuche her.

6.9.186. Datei "/modules/pdf/class.webservices.php"

Funktionen in der Datei "/modules/pdf/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.187. Datei "/modules/pdf/container.php"

Funktionen in der Datei "/modules/pdf/container.php" (Name der Funktion, Beschreibung, Parameter):

pdf_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="pdf" [...]}	$form

6.9.188. Datei "/modules/poll/class.admin.php"

Funktionen in der Datei "/modules/poll/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_poll()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_poll()	Fügt eine neue Umfrage zur Datenbank hinzu.	$form
do_delete_poll()	Löscht eine bestehende Umfrage samt der abgegebenen Stimmen aus der Datenbank.	$number
do_edit_poll()	Zeigt das Formular für die Bearbeitung einer bestehenden Umfrage an.	$number
do_edit_poll_2()	Aktualisiert die Angaben zu einer Umfrage in der Datenbank.	$form
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.189. Datei "/modules/poll/class.base.php"

Funktionen in der Datei "/modules/poll/class.base.php" (Name der Funktion, Beschreibung, Parameter):

show_poll()	Zeigt je nach übergebenen Parametern die passende Umfrage an.	$type, $width = "1", $mode = "0", $show_mode = "0"
show_poll_free()	Zeit die Umfragen im Administrationsbereich ohne Einschränkungen an.	$number, $width = "1", $form = array()
show_poll_result()	Zeigt die Übersicht über eine Umfrage an, nachdem abgestimmt wurde.	$number
show_poll_vote()	Zeigt die Auswahl der möglichen Antworten für eine Umfrage an.	$number

6.9.190. Datei "/modules/poll/class.index.php"

Funktionen in der Datei "/modules/poll/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_poll()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_entry()	Zeigt abhängig vom Status des Benutzers eine einzelne Umfrage an.	$number
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	
do_vote()	Speichert eine Stimme eines Benutzers zu einer bestimmten Umfrage.	$number, $form
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.191. Datei "/modules/poll/class.webservices.php"

Funktionen in der Datei "/modules/poll/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.192. Datei "/modules/poll/container.php"

Funktionen in der Datei "/modules/poll/container.php" (Name der Funktion, Beschreibung, Parameter):

poll_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Dies umfasst u.a. den Container für die Anzeige einer beliebigen Umfrage. Aufruf in den Vorlagen: {cp:container module="poll" [...]}	$forn

6.9.193. Datei "/modules/search/class.admin.php"

Funktionen in der Datei "/modules/search/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_search ()	Konstruktur, dient der Initialisierung der Klasse und einiger Variablen.
install()	SQL- und Dateianweisungen zur Installation des Modules.
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.

6.9.194. Datei "/modules/search/class.base.php"

Funktionen in der Datei "/modules/search/class.base.php" (Name der Funktion, Beschreibung, Parameter):

do_wordlist()	Dies ist eine Liste mit Buchstaben, welche bei der Volltextsuche umgewandelt werden. D.h. es wird dann auch nach den jeweiligen Alternativen gesucht.

6.9.195. Datei "/modules/search/class.index.php"

Funktionen in der Datei "/modules/search/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_search ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_main()	Erzeugt die Darstellung der Suche mit den entsprechenden Ergebnissen.	$q, $category, $type, $live, $form
do_search()	Leitet eine Suchabfrage auf die eigentliche Suchseite um.	$q, $category, $type, $live, $form
do_stopwords()	In dieser Liste sind häufig genutzte Wörter enthalten, welche bei der Volltextsuche nicht beachtet werden sollen. Es können hier beliebige weitere Wörter hinzugefügt werden.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.196. Datei "/modules/search/class.webservices.php"

Funktionen in der Datei "/modules/search/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.197. Datei "/modules/search/container.php"

Funktionen in der Datei "/modules/search/container.php" (Name der Funktion, Beschreibung, Parameter):

search_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Für die Volltextsuche ist dies lediglich die Darstellung des Suchformulares. Aufruf in den Vorlagen: {cp:container module="search" [template="vorlage.html"]}	$form

6.9.198. Datei "/modules/shop/alternative_price.php"

Funktionen in der Datei "/modules/shop/alternative_price.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_alternative_price ()	Fügt die neue Preisstaffel für einen Artikel ein.	$form, $number
shop_delete_alternative_price ()	Löscht die Preisstaffel für einen Artikel komplett.	$number

6.9.199. Datei "/modules/shop/class.admin.php"

Funktionen in der Datei "/modules/shop/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_shop ()	Konstruktur, dient der Initialisierung der Klasse und einiger Variablen.	
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	
do_shortnavigation()	Zeigt die Kurznavigation auf der linken Seite an.	
install()	SQL- und Dateianweisungen zur Installation des Modules.	
shop_error_message()	Gibt eine Fehlermeldung an den Benutzer aus.	$number
shop_items_sold()	Liefert die Anzahl der bisherigen Verkäufe eines Artikels zurück.	$number
skrip_test()	Führt einen internen Funktionstest durch.	$amount
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.200. Datei "/modules/shop/class.clients.php"

Funktionen in der Datei "/modules/shop/class.clients.php" (Name der Funktion, Beschreibung, Parameter):

delete_client()	Löscht einen bzw. mehrere Kunden über die Kundennummer.	$number
get_client()	Liefert alle Felder bzw. ein bestimmtes Feld zu einem Kunden anhand der Kundennummer.	$number, $field = ""

6.9.201. Datei "/modules/shop/class.exchange.php"

Funktionen in der Datei "/modules/shop/class.exchange.php" (Name der Funktion, Beschreibung, Parameter):

add_exchange()	Fügt eine neue Import-/Exportdefinition zur Datenbank hinzu.	$type, $section, $format, $title, $array
get_exchange()	Liefert eine Import-/Exportdefinition als Array zurück.	$number
get_exchange_array()	Liefert die verfügbaren Import-/Exportdefinitionen für einen Bereich als Array zurück.	$type, $section

6.9.202. Datei "/modules/shop/class.index.php"

Funktionen in der Datei "/modules/shop/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_shop ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	
frontend_editing()	Behandelt die Einblendung der Frontend-Editing-Leiste für berechtigte Benutzer.	
metatags()	Behandelt die Einblendung der Meta-Tags.	
search()	Stellt die Verbindung zur Volltextsuche her.	
shop_cart_message()	Liefert eine Meldung aus dem Warenkorb an den Benutzer zurück.	$query, $target = ""
shop_maintenance()	Zeigt eine Meldung an, wenn sich der Shop im Wartungsmodus befindet.	
shop_replace()	Zentrale Funktion, die die Verbindung des Shop-Layouts mit dem Inhalt herstellt.	$site, $cp_content

6.9.203. Datei "/modules/shop/class.reviews.php"

Funktionen in der Datei "/modules/shop/class.reviews.php" (Name der Funktion, Beschreibung, Parameter):

show_review()	Liefert eine fertig formatierte Kundenmeinung nach einer Vorlage zurück.	$template, $array

6.9.204. Datei "/modules/shop/class.statistic.php"

Funktionen in der Datei "/modules/shop/class.statistic.php" (Name der Funktion, Beschreibung, Parameter):

get_average_amount()	Liefert den durchschnittlichen Umsatz pro Bestellung (unformatiert).	
get_cancellations_amount	Liefert den Umsatzverlust durch Stornierungen zurück (unformatiert).	

()		
get_cart_items()	Liefert die Anzahl der Artikel im Warenkorb zurück.	
get_cart_user()	Liefert die Gesamtzahl der Besucher zurück, welche Artikel im Warenkorb haben.	
get_num_clients()	Liefert die Anzahl der registrierten Kunden zurück.	
get_num_diff_clients()	Liefert die Anzahl an Bestellungen, die von unterschiedlichen Kunden stammen.	
get_num_hits()	Liefert die Anzahl der bisherigen Hits auf alle Artikel zurück.	
get_num_order_items()	Liefert die gesamte Anzahl an bestellten Artikel zurück.	$time_start, $time_stop = ""
get_num_orders()	Liefert die gesamte Anzahl an Bestellungen in einem Zeitraum zurück.	$time_start, $time_stop = "", $array = array()
get_total_amount()	Liefert den Gesamtumsatz zurück (unformatiert).	

6.9.205. Datei "/modules/shop/class.webservices.php"

Funktionen in der Datei "/modules/shop/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

get_cart()	Simuliert Warenkorb über Webservices.	$form
get_items()	Liefert Informationen über die Artikel.	$form, $search_string
get_manufacturer()	Liefert Informationen über einen Hersteller und dessen Produkte.	$form
get_prosearch()	Sucht die passenden Artikel über boolsche Verknüpfungen usw.	$form
get_reviews()	Liefert die Kundenmeinungen zu einer Artikelnummer / ID.	$form
get_rssfeed()	Liefert Informationen über die neuesten Artikel aus RSS-Feed zurück. Aufruf: http://www.domain.de/pfad/webservices.php?file=shop&rssfeed=yes&(company|keyword|limit|section)=...	$form
get_rssfeed_cdata()	Säubert die Angaben im CDATA-Elemente und schickt diese wieder zurück.	$string, $cdata = "1"
get_sections()	Liefert Informationen über die Kategorien.	$form
get_similar()	Sucht ähnliche Artikel und liefert deren Informationen.	$form
get_single_item()	Fügt den Datensatz für einen Artikel hinzu.	$row, $select_array
get_wishlist()	Sucht die Artikel in einer Wunschliste.	$form
process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.206. Datei "/modules/shop/container.php"

Funktionen in der Datei "/modules/shop/container.php" (Name der Funktion, Beschreibung, Parameter):

shop_container()	Zentrale Zusammenstellung der Container für die Ausgabe auf der Website. Aufruf in den Vorlagen: {cp:container module="shop" type="..." [...]}	$form

6.9.207. Datei "/modules/shop/count.php"

Funktionen in der Datei "/modules/shop/count.php" (Name der Funktion, Beschreibung, Parameter):

shop_category_views()	Zählt die Anzahl der Zugriffe auf die Artikel einer Kategorie (samt Unterkategorien).	$number, $count = "0"

6.9.208. Datei "/modules/shop/currency.php"

Funktionen in der Datei "/modules/shop/currency.php" (Name der Funktion, Beschreibung, Parameter):

shop_currency_choose()	Wechselt die Währung des aktuellen Benutzers im Shop.	$code
shop_currency_export()	Formatiert eine Währungsausgabe aus der Datenbank für den Export ("nnnn.nn").	$string
shop_currency_replace ()	Tauscht in der Währungsangabe "." durch "," aus.	$string, $type = ""
shop_currency_shorten ()	Formatiert eine Währungsausgabe aus der Datenbank in das übliche Format ("n.nnn,nn").	$string

6.9.209. Datei "/modules/shop/misc.php"

Funktionen in der Datei "/modules/shop/misc.php" (Name der Funktion, Beschreibung, Parameter):

shop_clean_cart()	Löscht Artikel aus dem Warenkorb, die länger als die max. Speicherzeit dort liegen.	
shop_create_userkey()	Erzeugt eine eindeutige Kennung für einen Besucher/Kunden/Partner, bereits vorhandene Kennungen für Kunden werden übersprungen.	
shop_db_config()	Aktualisiert die Einstellungen in der Datei "/modules/shop/config.db.php" (Währungen, Steuersätze, Lieferstati).	$db_tables = ""
shop_delete_item_frontend ()	Löscht einen Artikel aus dem Shop inkl. aller Mediendateien, Kundenmeinungen etc.	$number
shop_filter_items()	Liefert eine SQL-Abfrage für die Filterung der Shop-Artikel auf allen öffentlich zugänglichen Seiten des Online-Shops zurück (u.a. Status, verfügbare Anzahl).	$show = "base", $prefix = "a.'
shop_get_altprices()	Liefert den Preis eines Artikels bei einer bestimmten Anzahl zurück (wenn für den Artikel	$number, $price, $amount

	eine Preisstaffel genutzt wird).	
shop_get_basetitle()	Gibt den kompletten Namen einer Variante zurück (wahlweise mit oder ohne Name des Basis-Artikels.	$number, $title = ""
shop_get_client_title()	Gibt Informationen über die verfügbaren Werte für den Titel / den akademischen Grad eines Kunden zurück.	$name = ""
shop_get_configuration ()	Gibt ein Array mit den erweiterten Einstellungen des Shops zurück.	$configuration = array()
shop_get_delivery()	Gibt Informationen über die verfügbaren Lieferstati als Array zurück.	$number = ""
shop_get_discountprices ()	Liefert anhand der Kundengruppe eines Kunden den Preis eines einzelnen Artikels abzüglich zutreffender Rabatte zurück.	$number, $price = "", $client = ""
shop_get_discountprices_sections ()	Liefert die Kategorienstruktur für einen Artikel für die Berechnung der Rabatte zurück.	$number, $array = array()
shop_get_file_path()	Erzeugt zentral die Pfade zum Shop, zu den Artikeln etc. innerhalb der PHP-Dateien.	$type, $array = array(), $dynamic = "0", $encode = "0"
shop_get_groupdiscount ()	Gibt in einem Array Informationen über die Kundengruppen-Rabatte für einen einzelnen Artikel zurück.	$number
shop_get_info()	Liefert abhängig von den Parametern Informationen zu verschiedenen Datenbanktabellen zurück.	$table, $number, $field = ""
shop_get_items()	Gibt Informationen über einzelne Artikel zurück.	$number, $field = ""
shop_get_multi_language ()	Gibt die mehrsprachigen Inhalte zu Zahlungs- und Versandarten, Artikeln und Kategorien zurück.	$number, $type, $data = array (), $field = ""
shop_get_order_source ()	Gibt Informationen über die verfügbaren Werte für den Bestelleingang der Bestellungen zurück.	$name = ""
shop_get_order_status()	Gibt Informationen über die individuellen Bestellstati als Array zurück.	$number = ""
shop_get_payment()	Gibt Informationen über eine Zahlungsart zurück.	$number, $field = ""
shop_get_sections()	Liefert ein Array mit Informationen zu den Kategorien eines Artikels zurück.	$number
shop_get_shipment()	Gibt Informationen über eine Versandart zurück.	$number, $field = ""
shop_get_subsections()	Ermittelt die Unterkategorien zu einer Kategorie.	$number, $array = array()
shop_get_template_path ()	Gibt in Abhängigkeit von den Einstellungen den Pfad zurück, unter dem die Vorlagen liegen.	
shop_get_tracking()	Gibt Informationen über das Tracking der Bestellung via DHL und Co. zurück.	$number = ""
shop_include_file()	Liefert den Inhalt einer vorhandenen Datei (z.B. einer Vorlage) zurück.	$filename
shop_item_salesrank()	Ermittelt den Verkaufsrang zu einem Artikel.	$number
shop_order_calc_prices ()	Liefert ein Array mit Preisinformationen zu einer Bestellung zurück (u.a. Warenwert, Steueranteil, Versand- und Zahlungskosten, Gutschein, Gesamtpreis).	$number
shop_order_number()	Formatiert die Bestellnummer abhängig von den übergebenen Angaben.	$string, $number, $date = ""
shop_select_taxrate()	Gibt Informationen über einen Steuersatz zurück.	$field = "rate", $number
shop_session_data()	Liefert die Session-Informationen des aktuellen Besuchers/Kunden/Partners zurück.	
shop_total_salesranks()	Zählt die maximale Anzahl der verfügbaren Verkaufsränge.	

6.9.210. Datei "/modules/shop/print_misc.php"

Funktionen in der Datei "/modules/shop/print_misc.php" (Name der Funktion, Beschreibung, Parameter):

shop_print_fax()	Ruft das aktuelle Artikelsortiment als FAX-Formular im PDF-Format auf.	
shop_print_labels()	Über diese Funktion werden die Etiketten für Bestellungen gedruckt.	$orders
shop_print_packaging()	Über diese Funktion werden die Packlisten gedruckt.	$orders

6.9.211. Datei "/modules/shop/print_order.php"

Funktionen in der Datei "/modules/shop/print_order.php" (Name der Funktion, Beschreibung, Parameter):

shop_print_order()	Über diese Funktion werden die Rechnungen im HTML- und Text-Format gedruckt.	$encode, $form, $return = ""

6.9.212. Datei "/modules/shop/router.php"

Funktionen in der Datei "/modules/shop/router.php" (Name der Funktion, Beschreibung, Parameter):

shop_email_shipment()	Diese Email informiert einen Kunden über den Versand seiner Bestellung.	$number, $redirect = "1"

6.9.213. Datei "/modules/shop/section_menu.php"

Funktionen in der Datei "/modules/shop/section_menu.php" (Name der Funktion, Beschreibung, Parameter):

shop_section_list()	Erzeugt für die Kategorienverwaltung eine Auflistung der Kategorien.	$under = "0", $count = "", $type = "", $block_array = array(), $max_count = "", $active_count = "0"
shop_section_menu()	Erzeugt für die Kategorienverwaltung eine Auflistung der Kategorien.	$under = "0", $count = "0", $selection = "", $this_admin = "1", $max_count = ""

6.9.214. Datei "/modules/shop/show_article.php"

Funktionen in der Datei "/modules/shop/show_article.php" (Name der Funktion, Beschreibung, Parameter):

shop_show_article()	Formatiert einen einzelnen Artikel für verschiedene Bereiche des Shops (u.a. auch die Druckansicht und die Ausgabe von Artikelinformationen in Emails).	$form

6.9.215. Datei "/modules/shop/admin/add_partner.php"

Funktionen in der Datei "/modules/shop/admin/add_partner.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_partner()	Zeigt das Formular zum Hinzufügen eines neuen Partner an.	$form
shop_add_partner_2()	Fügt einen neuen Partner zur Datenbank hinzu.	$form

6.9.216. Datei "/modules/shop/admin/article.php"

Funktionen in der Datei "/modules/shop/admin/article.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_article()	Fügt einen neuen Artikel zur Datenbank hinzu.	$form, $itemimage
shop_article()	Anzeige des Formulares zum Hinzufügen / Bearbeiten von Artikeln.	$number, $type
shop_blocked_article()	Zeigt die Information über einen geblockten (d.h. derzeit in Bearbeitung befindlichen) Artikel an.	$number, $type, $form = array(
shop_delete_article()	Löscht einen vorhandenen Artikel aus der Datenbank.	$number
shop_duplicate_article()	Kopiert einen vorhandenen Artikel inkl. aller damit verknüpften Mediendateien, Bewertungen etc.	$number, $redirect = "1"
shop_edit_article()	Aktualisiert die Angaben zu einem Artikel in der Datenbank.	$form, $itemimage
shop_edit_item_lang()	Formular für die Bearbeitung der fremdsprachigen Inhalte eines vorhandenen Artikels.	$form
shop_edit_item_lang_2()	Aktualisiert die fremdsprachigen Inhalte eines vorhandenen Artikels.	$form

6.9.217. Datei "/modules/shop/admin/article_crossselling.php"

Funktionen in der Datei "/modules/shop/admin/article_crossselling.php" (Name der Funktion, Beschreibung, Parameter):

shop_article_crossselling()	Auflistung der Cross-Selling-Artikel für einen Artikel.	$number
shop_article_crossselling_add()	Fügt für einen Artikel einen neuen Cross-Selling-Artikel hinzu.	$number, $form
shop_article_crossselling_delete()	Löscht einen vorhandenen Cross-Selling-Artikel eines Artikels.	$number, $selected, $type

6.9.218. Datei "/modules/shop/admin/article_media.php"

Funktionen in der Datei "/modules/shop/admin/article_media.php" (Name der Funktion, Beschreibung, Parameter):

shop_article_media()	Auflistung der Mediendateien für einen Artikel.	$number
shop_article_media_2()	Lädt neue Mediendateien für einen Artikel hoch.	$number, $form
shop_article_media_delete()	Löscht eine vorhandene Mediendatei eines Artikels.	$number, $selected, $type
shop_article_media_edit()	Bearbeitung der Mediendateien für einen Artikel.	$number, $selected
shop_article_media_edit_2()	Aktualisiert die Mediendateien für einen Artikel in der Datenbank.	$form

6.9.219. Datei "/modules/shop/admin/article_owner_rating.php"

Funktionen in der Datei "/modules/shop/admin/article_owner_rating.php" (Name der Funktion, Beschreibung, Parameter):

shop_article_owner_rating()	Bearbeitung der Bewertung eines Artikels durch den Shop-Betreiber.	$number
shop_article_owner_rating_2()	Speichert die Bewertung eines Artikels durch den Shop-Betreiber.	$number, $form

6.9.220. Datei "/modules/shop/admin/article_prices.php"

Funktionen in der Datei "/modules/shop/admin/article_prices.php" (Name der Funktion, Beschreibung, Parameter):

shop_article_prices()	Bearbeitung der Preisstaffelung für einen Artikel.	$number

6.9.221. Datei "/modules/shop/admin/article_reviews.php"

Funktionen in der Datei "/modules/shop/admin/article_reviews.php" (Name der Funktion, Beschreibung, Parameter):

shop_article_reviews()	Bearbeitung der Kundenmeinungen für einen Artikel.	$number, $selected
shop_article_reviews_2 ()	Aktualisierung einer Kundenmeinung für einen Artikel.	$form
shop_article_reviews_delete ()	Löscht eine vorhandenen Kundenmeinung komplett aus der Datenbank.	$number, $selected

6.9.222. Datei "/modules/shop/admin/articles_lists.php"

Funktionen in der Datei "/modules/shop/admin/articles_lists.php" (Name der Funktion, Beschreibung, Parameter):

shop_articles_lists()	Generiert die unterschiedlichen Listen anhand des übergebenen SQL-Befehls.	$type, $form

6.9.223. Datei "/modules/shop/admin/attributes.php"

Funktionen in der Datei "/modules/shop/admin/attributes.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_attributes_group ()	Fügt eine neue Artikelklasse hinzu.	$form
shop_deactivate_attributes_group ()	Entfernt die Zuordnungen von Artikeln zu einer bestimmten Artikelklasse.	$number
shop_delete_attribute_file ()	Löscht die zugeordnete Datei eines Attributes eines Artikels.	$attribute, $number
shop_delete_attributes_group ()	Entfernt die Zuordnung der Artikelklasse samt Attributen vom Artikel.	$number
shop_edit_attributes_group ()	Formular für das Bearbeiten einer vorhandenen Artikelklasse.	$number
shop_edit_attributes_group_2 ()	Aktualisiert die Angaben zu einer vorhandenen Artikelklasse in der Datenbank.	$form
shop_group_attributes()	Hauptseite mit der Übersicht über die vorhandenen Artikelklassen und einem Formular zum Hinzufügen.	$form
shop_group_form()	Generiert das Formular zum Hinzufügen / Bearbeiten einer Artikelklasse.	$type, $array = array ()
shop_set_attribute_group ()	Legt die Artikelklasse für einen Artikel fest.	$form
shop_set_attribute_values ()	Fügt die Attribute für die Artikelklasse eines Artikels ein.	$form

6.9.224. Datei "/modules/shop/admin/change_commission.php"

Funktionen in der Datei "/modules/shop/admin/change_commission.php" (Name der Funktion, Beschreibung, Parameter):

shop_change_commission ()	Ändert den Status einer Provision eines Partners (nicht bezahlt / bezahlt).	$number, $partner
shop_delete_commission ()	Löscht eine vorhandene Provision eines Partners (z.B. bei Stornierung der Bestellung).	$number, $partner

6.9.225. Datei "/modules/shop/admin/check.php"

Funktionen in der Datei "/modules/shop/admin/check.php" (Name der Funktion, Beschreibung, Parameter):

shop_check()	Überprüft ausländische UmSt.-ID's auf deren Gültigkeit.	$type, $form

6.9.226. Datei "/modules/shop/admin/choose_articles.php"

Funktionen in der Datei "/modules/shop/admin/choose_articles.php" (Name der Funktion, Beschreibung, Parameter):

shop_choose_articles()	Führt Aktionen wie Aktivieren/Deaktivieren gleichzeitig für mehrere Artikel aus.	$setaction, $checkbox
shop_choose_articles_options ()	Stellt die Optionen für das Dropdown in der Übersicht der Artikel bereit.	

6.9.227. Datei "/modules/shop/admin/choose_clients.php"

Funktionen in der Datei "/modules/shop/admin/choose_clients.php" (Name der Funktion, Beschreibung, Parameter):

shop_choose_clients()	Führt Aktionen wie Aktivieren/Deaktivieren gleichzeitig für mehrere Kunden aus.	$setaction, $checkbox
shop_choose_clients_options ()	Stellt die Optionen für das Dropdown in der Übersicht der Kunden bereit.	

6.9.228. Datei "/modules/shop/admin/choose_companies.php"

Funktionen in der Datei "/modules/shop/admin/choose_companies.php" (Name der Funktion, Beschreibung, Parameter):

shop_choose_companies ()	Führt Aktionen wie Aktivieren/Deaktivieren gleichzeitig für mehrere Hersteller aus.	$setaction, $checkbox
shop_choose_companies_options ()	Stellt die Optionen für das Dropdown in der Übersicht der Hersteller bereit.	

6.9.229. Datei "/modules/shop/admin/choose_orders.php"

Funktionen in der Datei "/modules/shop/admin/choose_orders.php" (Name der Funktion, Beschreibung, Parameter):

shop_choose_orders()	Führt Aktionen wie Aktivieren/Deaktivieren gleichzeitig für mehrere Bestellungen aus.	$setaction, $checkbox
shop_choose_orders_options ()	Stellt die Optionen für das Dropdown in der Übersicht der Bestellungen bereit.	

6.9.230. Datei "/modules/shop/admin/choose_partner.php"

Funktionen in der Datei "/modules/shop/admin/choose_partner.php" (Name der Funktion, Beschreibung, Parameter):

shop_choose_partner()	Führt Aktionen wie Aktivieren/Deaktivieren gleichzeitig für mehrere Partner aus.	$setaction, $checkbox
shop_choose_partner_options ()	Stellt die Optionen für das Dropdown in der Übersicht der Partner bereit.	

6.9.231. Datei "/modules/shop/admin/choose_reviews.php"

Funktionen in der Datei "/modules/shop/admin/choose_reviews.php" (Name der Funktion, Beschreibung, Parameter):

shop_choose_reviews()	Führt Aktionen wie Aktivieren/Deaktivieren gleichzeitig für mehrere Kundenmeinungen aus.	$setaction, $checkbox

6.9.232. Datei "/modules/shop/admin/client.php"

Funktionen in der Datei "/modules/shop/admin/client.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_client()	Formular zum Hinzufügen eines neuen Kunden.	
shop_add_client_2()	Fügt einen neuen Kunden in die Datenbank ein.	$form
shop_client_form()	Formular zum Hinzufügen / Bearbeiten eines Kunden.	$type, $number = ""
shop_edit_client()	Bearbeitung eines vorhandenen Kunden (u.a. mit Zusatzinformationen über die bisherigen Bestellungen und eine Statistik).	$number
shop_edit_client_2()	Aktualisiert die Daten eines vorhandenen Kunden in der Datenbank.	$form

6.9.233. Datei "/modules/shop/admin/client_comment.php"

Funktionen in der Datei "/modules/shop/admin/client_comment.php" (Name der Funktion, Beschreibung, Parameter):

shop_client_comment()	Formular für das Hinzufügen / Bearbeiten eines Kommentares zu einem Kunden.	$number
shop_client_comment_2 ()	Aktualisiert einen Kommentar zu einem Kunden (nur intern sichtbar).	$form

6.9.234. Datei "/modules/shop/admin/client_solvency.php"

Funktionen in der Datei "/modules/shop/admin/client_solvency.php" (Name der Funktion, Beschreibung, Parameter):

shop_client_solvency()	Überprüft die Bonität eines Kunden und liefert diese Informationen zurück.	$number, $client_array = array()
shop_client_solvency_details ()	Zeigt die Details der jeweils letzten Bonitätsprüfung eines Kunden an.	$number

6.9.235. Datei "/modules/shop/admin/crm.php"

Funktionen in der Datei "/modules/shop/admin/crm.php" (Name der Funktion, Beschreibung, Parameter):

shop_crm_delete()	Löscht eine Nachricht eines Kunden aus dem System.	$number
shop_crm_edit()	Zeigt ein Formular für die Bearbeitung einer einzelnen Nachricht an.	$number
shop_crm_edit_2()	Aktualisiert die Angaben zu einer einzelnen Nachricht in der Datenbank.	$form
shop_crm_import()	Importiert die neu im Ordner "modules/shop/import/crm/" vorhandenen Emails.	
shop_crm_import_email ()	Importiert eine einzelne Email auf der Basis des Dateinamens.	$filename
shop_crm_import_file()	Speichert die übermittelte Datei im Import-Ordner und startet den Import.	$form
shop_crm_quotedprintable ()	Dekodiert als Quoted-Printable kodierte Email-Adressen und Texte.	$string
shop_crm_show()	Zeigt den Nachrichtenverlauf mit einem einzelnen Kunden in einer Übersicht an.	$number, $form

shop_crm_show_messages ()	Zeigt den Nachrichtenverlauf mit einem einzelnen Kunden im Detail an.	$number
shop_crm_show_now()	Verarbeitet die Parameter der Suche und leitet auf die Ergebnisseite um.	$form

6.9.236. Datei "/modules/shop/admin/delete_order.php"

Funktionen in der Datei "/modules/shop/admin/delete_order.php" (Name der Funktion, Beschreibung, Parameter):

shop_delete_order()	Entfernt eine einzelne Bestellung samt Artikeln aus der Datenbank.	$number
shop_delete_orders()	Entfernt mehrere Bestellungen samt Artikeln aus der Datenbank.	$orders

6.9.237. Datei "/modules/shop/admin/delete_order_item.php"

Funktionen in der Datei "/modules/shop/admin/delete_order_item.php" (Name der Funktion, Beschreibung, Parameter):

shop_delete_order_item ()	Löscht einen einzelnen Artikel aus der Bestellung eines Kunden.	$form

6.9.238. Datei "/modules/shop/admin/delete_partner.php"

Funktionen in der Datei "/modules/shop/admin/delete_partner.php" (Name der Funktion, Beschreibung, Parameter):

shop_delete_partner()	Löscht einen vorhandenen Partner samt dessen Bestellungen aus der Datenbank.	$number, $return = "1"

6.9.239. Datei "/modules/shop/admin/edit_partner.php"

Funktionen in der Datei "/modules/shop/admin/edit_partner.php" (Name der Funktion, Beschreibung, Parameter):

shop_edit_partner()	Formular für die Bearbeitung eines vorhandenen Partners.	$number
shop_edit_partner_2()	Aktualisiert die Angaben zu einem Partner in der Datenbank.	$form
shop_show_partner_clicks ()	Zeigt die bisherigen Klicks durch die Besucher dieses Partners in einem Iframe an.	$number

6.9.240. Datei "/modules/shop/admin/export.php"

Funktionen in der Datei "/modules/shop/admin/export.php" (Name der Funktion, Beschreibung, Parameter):

shop_export()	Initialisiert den Export für die Artikel, Kunden und Bestellungen.	$form
shop_export_bmecat()	Generiert die gewünschten Daten im BMEcat-Format.	$form
shop_export_csv()	Generiert die gewünschten Daten im CSV-Format.	$form
shop_export_dta()	Generiert die gewünschten Daten für das Datenträgeraustausch-Verfahren (DTA).	$form
shop_export_dta_fill()	Hilfsfunktion für die Generierung von Daten im DTA-Format.	$max, $string = ""
shop_export_dta_format ()	Hilfsfunktion für die Generierung von Daten im DTA-Format.	$string
shop_export_dta_spacer ()	Hilfsfunktion für die Generierung von Daten im DTA-Format.	$max, $string = ""
shop_export_excel()	Generiert die gewünschten Daten im Excel-CSV-Format.	$form
shop_export_extras()	Holt zusätzliche Daten für den Export und formatiert die Werte.	$table, $array
shop_export_file()	Liefert die generierten Daten auf dem gewünschten Weg aus.	$form, $content, $filetype = "text/csv", $filename = "shop_data.csv"
shop_export_sql()	Generiert die gewünschten Daten im SQL-Format.	$form
shop_export_xml()	Generiert die gewünschten Daten im XML-Format.	$form

6.9.241. Datei "/modules/shop/admin/export_articles.php"

Funktionen in der Datei "/modules/shop/admin/export_articles.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_articles()	Anzeige des Artikel-Exports mit der Auswahl des gewünschten Formates etc.	$form
shop_export_get_articles ()	Liest die Liste der verfügbaren alternativen Exportformate.	

6.9.242. Datei "/modules/shop/admin/export_clients.php"

Funktionen in der Datei "/modules/shop/admin/export_clients.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_clients()	Anzeige des Artikel-Exports mit der Auswahl des gewünschten Formates etc.	$form
shop_export_get_clients ()	Liest die Liste der verfügbaren alternativen Exportformate.	

6.9.243. Datei "/modules/shop/admin/export_orders.php"

Funktionen in der Datei "/modules/shop/admin/export_orders.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_get_orders ()	Liest die Liste der verfügbaren alternativen Exportformate.	
shop_export_orders()	Anzeige des Artikel-Exports mit der Auswahl des gewünschten Formates etc.	$form

6.9.244. Datei "/modules/shop/admin/export_sections.php"

Funktionen in der Datei "/modules/shop/admin/export_sections.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_get_sections ()	Liest die Liste der verfügbaren alternativen Exportformate.	
shop_export_sections()	Anzeige des Kategorie-Exports mit der Auswahl des gewünschten Formates etc.	$form

6.9.245. Datei "/modules/shop/admin/export_vcard.php"

Funktionen in der Datei "/modules/shop/admin/export_vcard.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_vcard()	Export die Kundendaten als vCard für die Einbindung in externe Email-Programme usw.	$number

6.9.246. Datei "/modules/shop/admin/extern_info.php"

Funktionen in der Datei "/modules/shop/admin/extern_info.php" (Name der Funktion, Beschreibung, Parameter):

shop_extern_footer()	Fussbereich der externen Information.	
shop_extern_header()	Kopfbereich der externen Information.	
shop_extern_info()	Zeigt den Rahmen für die Einbettung von umfangreicheren Information / Anleitungen an.	$type

6.9.247. Datei "/modules/shop/admin/import_articles.php"

Funktionen in der Datei "/modules/shop/admin/import_articles.php" (Name der Funktion, Beschreibung, Parameter):

shop_import_articles()	Formular für den Import von Artikeldaten aus externen Dateien.	$form
shop_import_articles_data_now ()	Führt den Import der Artikeldaten aus externen Dateien durch.	$form, $options
shop_import_articles_file ()	Liefert die URL für die Importdatei zurück.	$file_name = ""
shop_import_articles_now ()	Führt den Import der Artikeldaten aus externen Dateien durch.	$form, $importfile
shop_import_get_company ()	Überprüft, ob ein Hersteller bereits existiert. Wenn nicht, dann wird dieser neu angelegt und der Artikel dem Hersteller zugeordnet.	$number, $company
shop_import_get_section ()	Überprüft, ob eine Kategorie bereits existiert. Wenn nicht, dann wird diese neu angelegt.	$number, $section

6.9.248. Datei "/modules/shop/admin/import_clients.php"

Funktionen in der Datei "/modules/shop/admin/import_clients.php" (Name der Funktion, Beschreibung, Parameter):

shop_import_clients()	Formular für den Import von Kundendaten aus externen Dateien.	$form
shop_import_clients_data_now ()	Führt den Import der Artikeldaten aus externen Dateien durch.	$form, $options
shop_import_clients_file ()	Liefert die URL für die Importdatei zurück.	$file_name
shop_import_clients_now ()	Führt den Import der Kundendaten aus externen Dateien durch.	$form, $importfile

6.9.249. Datei "/modules/shop/admin/import_orders.php"

Funktionen in der Datei "/modules/shop/admin/import_orders.php" (Name der Funktion, Beschreibung, Parameter):

shop_import_orders()	Formular für den Import von Bestellungen und Zahlungseingängen aus externen Dateien.	$form
shop_import_orders_now ()	Führt den Import der Bestellungen und Zahlungseingänge aus externen Dateien durch.	$form, $importfile

6.9.250. Datei "/modules/shop/admin/import_sections.php"

Funktionen in der Datei "/modules/shop/admin/import_sections.php" (Name der Funktion, Beschreibung, Parameter):

shop_import_sections()	Formular für den Import von Kategoriedaten aus externen Dateien.	$form
shop_import_sections_data_now ()	Führt den Import der Kategoriedaten aus externen Dateien durch.	$form, $options
shop_import_sections_now ()	Führt den Import der Kategoriedaten aus externen Dateien durch.	$form, $importfile

6.9.251. Datei "/modules/shop/admin/list_articles.php"

Funktionen in der Datei "/modules/shop/admin/list_articles.php" (Name der Funktion, Beschreibung, Parameter):

shop_list_article_sections ()	Generiert die Liste der Kategorien für die eingeschränkte Anzeige der Artikel.	$under, $string, $selection, $count = "0"
shop_list_articles()	Hauptseite für die Artikel mit der Auflistung der vorhandenen Artikel und allen Optionen.	$form
shop_list_articles_fields ()	Aktualisiert die Auswahl der Spalten in der Übersicht der Artikel.	$form

6.9.252. Datei "/modules/shop/admin/list_clients.php"

Funktionen in der Datei "/modules/shop/admin/list_clients.php" (Name der Funktion, Beschreibung, Parameter):

shop_list_clients()	Hauptseite für die Kunden mit der Auflistung der vorhandenen Kunden und allen Optionen.	$form
shop_list_clients_content ()	Stellt den Inhalt für die Hauptseite der Kundenauflistung bereit.	$form, $field_array
shop_list_clients_fields()	Aktualisiert die Auswahl der Spalten in der Übersicht der Kunden.	$form

6.9.253. Datei "/modules/shop/admin/list_coupons.php"

Funktionen in der Datei "/modules/shop/admin/list_coupons.php" (Name der Funktion, Beschreibung, Parameter):

shop_activate_coupons ()	Setzt den Status der Gutscheine auf "Aktiviert".	$array
shop_add_coupon()	Fügt einen neuen Gutschein zur Datenbank hinzu.	$form
shop_clean_coupons()	Setzt den Status der Gutscheine auf "Deaktiviert".	$array
shop_delete_coupon()	Entfernt einen in der Datenbank vorhandenen Gutschein.	$number
shop_delete_coupons()	Entfernt alle ausgewählten Wunschzettel aus der Datenbank.	$array
shop_edit_coupon()	Zeigt das Formular zur Bearbeitung / Überprüfung eines Gutscheines an.	$number
shop_edit_coupon_2()	Aktualisiert die Angaben zu einem Gutschein in der Datenbank.	$number, $form
shop_export_coupons()	Gibt die noch nicht genutzten, öffentlichen Gutscheincodes in einer kommaseparierten Liste aus.	
shop_list_coupons()	Hauptseite für die Erstellung und Verwaltung von Gutscheinen.	$form

6.9.254. Datei "/modules/shop/admin/list_groups.php"

Funktionen in der Datei "/modules/shop/admin/list_groups.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_group()	Fügt eine neue Kundengruppe in die Datenbank ein.	$form
shop_delete_group()	Löscht eine vorhandene Kundengruppe, sofern dieser keine Kunden zugewiesen sind.	$number
shop_edit_group()	Formular für die Bearbeitung einer vorhandenen Kundengruppe.	$number
shop_edit_group_2()	Aktualisiert die Angaben einer vorhandenen Kundengruppe.	$form
shop_group_discounts()	Speichert die Rabatte für eine Kundengruppe in der Datenbank ab.	$form
shop_list_groups()	Allgemeine Übersicht über die bereits vorhandenen Kundengruppen.	$form
shop_menu_groups()	Generiert ein Auswahlmenü der Kategorien für den Bearbeiten- / Löschen-Dialog.	$level, $count = "0", $active = "0", $base = "0", $blocked
shop_menu_items()	Generiert ein Auswahlmenü der Artikel für den Bearbeiten- / Löschen-Dialog.	$level, $count = "0", $active = "0", $base = "0", $blocked
shop_reset_group()	Setzt die Kundengruppe zurück, indem alle zugeordneten Kunden entfernt werden.	$number

6.9.255. Datei "/modules/shop/admin/list_orders.php"

Funktionen in der Datei "/modules/shop/admin/list_orders.php" (Name der Funktion, Beschreibung, Parameter):

shop_list_orders()	Hauptseite für die Bestellungen mit der Auflistung der vorhandenen Bestellungen und allen Optionen.	$form
shop_list_orders_fields()	Aktualisiert die Auswahl der Spalten in der Übersicht der Bestellungen.	$form

6.9.256. Datei "/modules/shop/admin/list_partner.php"

Funktionen in der Datei "/modules/shop/admin/list_partner.php" (Name der Funktion, Beschreibung, Parameter):

shop_list_partner()	Allgemeine Übersicht über die bereits vorhandenen Partner im Partnerprogramm.	$form
shop_list_partner_content ()	Stellt den Inhalt für die Hauptseite der Partnerauflistung bereit.	$form, $field_array
shop_list_partner_fields ()	Aktualisiert die Auswahl der Spalten in der Übersicht der Partner.	$form

6.9.257. Datei "/modules/shop/admin/list_wishlists.php"

Funktionen in der Datei "/modules/shop/admin/list_wishlists.php" (Name der Funktion, Beschreibung, Parameter):

shop_list_wishlists()	Hauptseite für die Überprüfung der Wunschzettel der Kunden.	$form

6.9.258. Datei "/modules/shop/admin/mail_clients.php"

Funktionen in der Datei "/modules/shop/admin/mail_clients.php" (Name der Funktion, Beschreibung, Parameter):

shop_mail_clients()	Formular für das Verfassen von Emails an Kunden.	$array, $string
shop_mail_clients_now ()	Versendet an einzelne / mehrere Kunden eine Information per Email.	$form

6.9.259. Datei "/modules/shop/admin/main.php"

Funktionen in der Datei "/modules/shop/admin/main.php" (Name der Funktion, Beschreibung, Parameter):

shop_main()	Startseite von CP::Shop mit der Auflistung der Hauptbereiche und den Unterseiten.	

6.9.260. Datei "/modules/shop/admin/misc.fields.php"

Funktionen in der Datei "/modules/shop/admin/misc.fields.php" (Name der Funktion, Beschreibung, Parameter):

shop_date_field()	Stellt ein Datumsfeld für die Attribute von Artikeln bereit.	$number, $content

6.9.261. Datei "/modules/shop/admin/options.php"

Funktionen in der Datei "/modules/shop/admin/options.php" (Name der Funktion, Beschreibung, Parameter):

shop_options()	Hauptansicht der Einstellungen mit den Formularen und der Auswahl der weiteren Optionen.	$form
shop_set_options()	Aktualisiert die allgemeinen Einstellungen in der Datenbank.	$form, $countries

6.9.262. Datei "/modules/shop/admin/options_companies.php"

Funktionen in der Datei "/modules/shop/admin/options_companies.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_company()	Fügt einen neuen Hersteller in die Datenbank ein.	$form, $logo
shop_deactivate_items()	Deaktiviert alle zugeordneten Artikel eines Herstellers.	$number
shop_delete_company()	Löscht einen vorhandenen Hersteller aus der Datenbank.	$number, $return = "1"
shop_edit_company()	Aktualisiert die Angaben eines vorhandenen Herstellers in der Datenbank.	$form, $logo
shop_edit_company_lang ()	Formular für die Bearbeitung der fremdsprachigen Inhalte eines vorhandenen Herstellers.	$form
shop_edit_company_lang_2 ()	Aktualisiert die fremdsprachigen Inhalte eines vorhandenen Herstellers.	$form
shop_list_companies_content ()	Stellt den Inhalt für die Hauptseite der Herstellerauflistung bereit.	$form, $field_array
shop_options_companies ()	Hauptansicht für die Herstellerverwaltung mit dem Formular für das Hinzufügen / Bearbeiten.	$form, $number
shop_options_companies_fields ()	Aktualisiert die Auswahl der Spalten in der Übersicht der Hersteller.	$form

6.9.263. Datei "/modules/shop/admin/options_currencies.php"

Funktionen in der Datei "/modules/shop/admin/options_currencies.php" (Name der Funktion, Beschreibung, Parameter):

shop_edit_currency()	Formular für die Bearbeitung einer Währung.	$number
shop_edit_currency_2()	Aktualisiert die Angaben zu einer Währung in der Datenbank.	$form
shop_options_currencies ()	Zeigt die Übersichtsseite zu den Währungen in der Verwaltung an.	$form
shop_options_currencies_update ()	Gleicht die Währungskurse des Systems mit den Referenzkursen der EZB ab.	
shop_set_options_currencies ()	Aktualisiert die Einstellungen zu den Währungen im System.	$form

shop_standard_currency ()	Legt eine Währung als Standard im Shop fest.	$number
show_show_currencies ()	Wechselt zwischen der Anzeige aller Währungen und der Währungen mit Wechselkursen.	$show

6.9.264. Datei "/modules/shop/admin/options_delivery.php"

Funktionen in der Datei "/modules/shop/admin/options_delivery.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_delivery()	Formular für die Bearbeitung der Lieferstati im Shop.	$form
shop_set_options_delivery ()	Aktualisiert die Einstellungen für die Lieferstati in der Datenbank.	$form

6.9.265. Datei "/modules/shop/admin/options_ekomi.php"

Funktionen in der Datei "/modules/shop/admin/options_ekomi.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_ekomi()	Formular für die Anpassung der Einstellungen für eKomi.	$form
shop_options_ekomi_export ()	Startet den Mailversand zu den Kunden und exportiert die Angaben zu eKomi.	
shop_set_options_ekomi ()	Speichert die geänderten Einstellungen für eKomi in der Datenbank ab.	$form

6.9.266. Datei "/modules/shop/admin/options_emails.php"

Funktionen in der Datei "/modules/shop/admin/options_emails.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_emails()	Formular für die Anpassung der Einstellungen für die Emails.	$form
shop_options_emails_array ()	Liefert das Array mit den Email-Vorlagen im CP::Shop zurück.	
shop_set_options_emails ()	Speichert die geänderten Einstellungen für die Emails in den Dateien ab.	$form

6.9.267. Datei "/modules/shop/admin/options_extpartner.php"

Funktionen in der Datei "/modules/shop/admin/options_extpartner.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_extpartner ()	Aktualisiert die Einstellungen für die externen Partnerprogramme in der Datenbank.	$form

6.9.268. Datei "/modules/shop/admin/options_filter.php"

Funktionen in der Datei "/modules/shop/admin/options_filter.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_filter()	Formular für die Eingabe der Email-Adressen und Adressen, welche bei der Anmeldung eines Kunden oder bei der Bestellung geblockt werden sollen.	$form
shop_options_filter_address ()	Zeigt das Formular für zum Hinzufügen / Bearbeiten einer Adresse in der Robinson-Liste an.	$form = array()
shop_set_options_filter ()	Aktualisiert die geblockten Email-Adressen in der Datenbank.	$form
shop_set_options_filter_addresses ()	Aktualisiert die geblockten Adressen in der Datenbank (Robinson-Liste).	$form

6.9.269. Datei "/modules/shop/admin/options_order_status.php"

Funktionen in der Datei "/modules/shop/admin/options_order_status.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_order_status ()	Formular für die Bearbeitung der Bestellstati im Shop.	$form
shop_options_order_status_templates ()	Generiert das Dropdown-Menü für die Auswahl der verfügbaren Email-Vorlagen.	$emails, $select = ""
shop_set_options_order_status ()	Aktualisiert die Einstellungen für die Bestellstati in der Datenbank.	$form

6.9.270. Datei "/modules/shop/admin/options_partner.php"

Funktionen in der Datei "/modules/shop/admin/options_partner.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_partner()	Formular für die Bearbeitung der Einstellungen für das Partnerprogramm.	$form
shop_set_options_partner ()	Aktualisiert die Einstellungen für das Partnerprogramm in der Datenbank.	$form

6.9.271. Datei "/modules/shop/admin/options_payment.php"

Funktionen in der Datei "/modules/shop/admin/options_payment.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_payment()	Formular für das Hinzufügen einer neuen Zahlungsart.	
shop_add_payment_2()	Fügt eine neue Zahlungsart zur Datenbank hinzu.	$form
shop_delete_payment()	Löscht eine vorhandene Zahlungsart aus der Datenbank, sofern diese noch nicht genutzt wurde.	$number
shop_edit_payment()	Formular für die Bearbeitung einer Zahlungsart (Felder sind abhängig von der Zahlungsart).	$number
shop_edit_payment_2()	Aktualisiert die Angaben zu einer Zahlungsart in der Datenbank.	$form
shop_edit_payment_lang()	Formular für die Bearbeitung der fremdsprachigen Inhalte einer vorhandenen Zahlungsart.	$form
shop_edit_payment_lang_2()	Aktualisiert die fremdsprachigen Inhalte einer vorhandenen Zahlungsart.	$form
shop_install_payment()	Installiert eine bislang noch nicht im Shop genutzte Zahlungsart.	$form
shop_options_payment()	Zeigt die Übersicht der im Shop verfügbaren Zahlungsarten an.	$form
shop_standard_payment()	Legt eine Zahlungsart als Standard im Shop fest.	$number
shop_standard_paymenttax()	Liefert den Standard-MwSt.-Satz für die Zahlungsarten zurück.	
shop_uninstall_payment()	Deinstalliert eine bereits im Shop vorhandene, aber noch nicht in Bestellungen genutzte Zahlungsart.	$form

6.9.272. Datei "/modules/shop/admin/options_pdfinvoice.php"

Funktionen in der Datei "/modules/shop/admin/options_pdfinvoice.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_pdfinvoice()	Aktualisiert die Einstellungen für die PDF-Rechnung und das PDF-Faxformular in der Datenbank.	$form
shop_set_options_pdfinvoice()	Aktualisiert die Einstellungen für die PDF-Rechnung und das PDF-Faxformular in der Datenbank.	$form

6.9.273. Datei "/modules/shop/admin/options_reset_shop.php"

Funktionen in der Datei "/modules/shop/admin/options_reset_shop.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_reset_shop()	Diese Funktion setzt den Shop komplett zurück. ACHTUNG: Es werden alle Kunden, Bestellungen, Artikel, Partner usw. gelöscht!	$form

6.9.274. Datei "/modules/shop/admin/options_reviews.php"

Funktionen in der Datei "/modules/shop/admin/options_reviews.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_reviews()	Formular für die Bearbeitung der Einstellungen für die Kundenmeinungen.	$form
shop_set_options_reviews()	Aktualisiert die Einstellungen für die Kundenmeinungen in der Datenbank.	$form

6.9.275. Datei "/modules/shop/admin/options_rowisoft.php"

Funktionen in der Datei "/modules/shop/admin/options_rowisoft.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_rowisoft()	Formular für die Anpassung der Einstellungen für Rowisoft blue.	$form
shop_options_rowisoft_ini()	Liefert den Inhalt der für Rowisoft blue erforderlichen Datei "sandobade_cpshop.ini" zurück.	
shop_set_options_rowisoft()	Speichert die geänderten Einstellungen für Rowisoft blue in der Datenbank ab.	$form

6.9.276. Datei "/modules/shop/admin/options_search.php"

Funktionen in der Datei "/modules/shop/admin/options_search.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_search()	Formular für die Bearbeitung der Einstellungen für die Volltextsuche.	$form
shop_set_options_search()	Aktualisiert die Einstellungen für die Volltextsuche in der Datenbank.	$form, $fields

6.9.277. Datei "/modules/shop/admin/options_shipment.php"

Funktionen in der Datei "/modules/shop/admin/options_shipment.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_shipment()	Formular für das Hinzufügen einer neuen Versandart.	
shop_add_shipment_2()	Fügt eine neue Versandart zur Datenbank hinzu.	$form
shop_delete_shipment()	Löscht eine vorhandene Versandart aus der Datenbank, sofern diese noch nicht genutzt wurde.	$number
shop_edit_shipment()	Formular für die Bearbeitung einer Versandart.	$number
shop_edit_shipment_2()	Aktualisiert die Angaben zu einer Versandart in der Datenbank.	$form
shop_edit_shipment_lang ()	Formular für die Bearbeitung der fremdsprachigen Inhalte einer vorhandenen Versandart.	$form
shop_edit_shipment_lang_2 ()	Aktualisiert die fremdsprachigen Inhalte einer vorhandenen Versandart.	$form
shop_options_shipment ()	Zeigt die Übersicht der im Shop verfügbaren Versandarten an.	$form
shop_standard_shipment ()	Legt eine Versandart als Standard im Shop fest.	$number
shop_standard_shipmenttax ()	Liefert den Standard-MwSt.-Satz für die Versandarten zurück.	

6.9.278. Datei "/modules/shop/admin/options_solvency.php"

Funktionen in der Datei "/modules/shop/admin/options_solvency.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_solvency ()	Aktualisiert die Einstellungen für die Schnittstellen zur Bonitätsprüfung in der Datenbank.	$form

6.9.279. Datei "/modules/shop/admin/options_taxrates.php"

Funktionen in der Datei "/modules/shop/admin/options_taxrates.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_taxrates()	Formular für die Bearbeitung der Steuersätze im Shop.	$form
shop_set_options_taxrates ()	Aktualisiert die Einstellungen für die Steuersätze in der Datenbank.	$form

6.9.280. Datei "/modules/shop/admin/options_templates.php"

Funktionen in der Datei "/modules/shop/admin/options_templates.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_templates ()	Formular für die Anpassung der Einstellungen für die Vorlagen.	$form
shop_set_options_templates ()	Speichert die geänderten Einstellungen für die Vorlagen in den Dateien ab.	$form

6.9.281. Datei "/modules/shop/admin/options_trustedshops.php"

Funktionen in der Datei "/modules/shop/admin/options_trustedshops.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_trustedshops ()	Formular für die Anpassung der Einstellungen für Trusted Shops.	$form
shop_set_options_trustedshops ()	Speichert die geänderten Einstellungen für Trusted Shops in der Datenbank ab.	$form

6.9.282. Datei "/modules/shop/admin/options_zones.php"

Funktionen in der Datei "/modules/shop/admin/options_zones.php" (Name der Funktion, Beschreibung, Parameter):

shop_options_zones()	Formular für die Bearbeitung der Einstellungen der Länder- und Zonen-Verwaltung.	$form
shop_set_options_zones ()	Aktualisiert die Einstellungen zu den Zonen in der Datenbank.	$form
shop_set_options_zones_map ()	Aktualisiert die Zuordnung der Länder zu den Zonen in der Datenbank.	$form
shop_show_zones()	Wechselt zwischen der Anzeige aller Zonen und der Anzeige der Länder einer bestimmten Zone.	$show

6.9.283. Datei "/modules/shop/admin/order.php"

Funktionen in der Datei "/modules/shop/admin/order.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_order()	Formular für das Hinzufügen einer neuen Bestellung.	
shop_add_order_2()	Fügt auf der Basis der übergebenen Angaben eine neue Bestellung in die Datenbank ein.	$form
shop_add_order_item()	Fügt einer Bestellung einen weiteren Artikel hinzu.	$form
shop_edit_order()	Formular für die Bearbeitung einer vorhandenen Bestellung.	$number
shop_edit_order_2()	Aktualisiert die Angaben einer vorhandenen Bestellung in der Datenbank.	$form
shop_edit_order_item()	Ändert den Preis eines bestimmten Artikels in einer Bestellung.	$form

shop_merge_order()	Formular für das Zusammenlegen mehrerer Bestellungen.	
shop_merge_order_2()	Ordnet die Artikel aus mehreren Bestellungen einer Basis-Bestellung zu.	$form
shop_send_order()	Schickt eine Email an den Kunden über den Versand seiner Bestellung.	$number

6.9.284. Datei "/modules/shop/admin/order_status.php"

Funktionen in der Datei "/modules/shop/admin/order_status.php" (Name der Funktion, Beschreibung, Parameter):

shop_order_status()	Verschickt eine Email an einen Kunden bei der Änderung des Status einer Bestellung.	$form

6.9.285. Datei "/modules/shop/admin/sections.php"

Funktionen in der Datei "/modules/shop/admin/sections.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_section()	Fügt eine neue Kategorie zur Datenbank hinzu.	$form
shop_change_status_section ()	Ändert den Status einer Kategorie auf verfügbar / geschützt.	$number
shop_delete_section()	Löscht eine vorhandene Kategorie (samt Unterkategorien).	$form
shop_duplicate_section ()	Dupliziert eine vorhandene Kategorie inklusive der Artikelzuordnungen.	$number
shop_edit_section()	Formular für die Bearbeitung einer vorhandenen Kategorie.	$form
shop_edit_section_2()	Aktualisiert die Angaben zu einer Kategorie in der Datenbank.	$form, $articledesc, $articletext
shop_edit_section_lang ()	Formular für die Bearbeitung der fremdsprachigen Inhalte einer vorhandenen Kategorie.	$form
shop_edit_section_lang_2 ()	Aktualisiert die fremdsprachigen Inhalte einer vorhandenen Kategorie.	$form
shop_menu_options()	Generiert ein Auswahlmenü der Kategorien für den Bearbeiten- / Löschen-Dialog.	$level, $count = "0", $active = "0", $base = "0"
shop_section_menu()	Erstellt die Struktur der Kategorien für das Fenster im unteren Bereich.	$level, $count = "0", $order = "position", $dir = "asc"
shop_section_move()	Verschiebt eine Kategorie innerhalb der Struktur nach oben oder unten.	$type, $number
shop_sections()	Hauptansicht der Kategorien mit Struktur auf der linken Seite und Formularen auf der rechten Seite.	$form
shop_show_sections()	Zeigt das Fenster im unteren Bereich mit der Auflistung der Kategorien an.	$form

6.9.286. Datei "/modules/shop/admin/statistic.php"

Funktionen in der Datei "/modules/shop/admin/statistic.php" (Name der Funktion, Beschreibung, Parameter):

shop_client_order()	Listet die Kunden mit den meisten Bestellungen auf.	
shop_fav_wishlist()	Listet die beliebtesten Wunschzettel auf.	
shop_fav_wishlist_item ()	Listet die auf den Wunschzetteln beliebtesten Artikel auf.	
shop_max_amount()	Listet die Kunden mit dem höchsten Umsatz auf.	
shop_max_wishlist()	Listet die umfangreichsten Wunschzettel auf (mit den meisten Artikeln).	
shop_statistic()	Hauptfunktion der Statistik mit allgemeiner Statistik und den Statistik-Reporten.	$form
shop_statistic_add_report ()	Fügt einen neuen Statistik-Report zum System hinzu.	$form
shop_statistic_arrays()	Liefert ein Array mit Angaben zu den verfügbaren Statistiken und Zeiträumen zurück.	$type = ""
shop_statistic_client_order ()	Ermittelt die Kunden mit den meisten Bestellungen.	$number = "", $form = array()
shop_statistic_delete_report ()	Löscht einen vorhandenen Statistik-Report aus der Datenbank.	$form
shop_statistic_edit_report ()	Formular für die Bearbeitung eines vorhandenen Statistik-Reports.	$form
shop_statistic_edit_report_2 ()	Aktualisiert die Angaben zu einem Statistik-Report in der Datenbank.	$form
shop_statistic_export_data ()	Exportiert die Daten als CSV-Datei.	$array
shop_statistic_fav_article ()	Liefert den nach der Anzahl der Bestellungen n-beliebtesten Artikel zurück.	$number = ""
shop_statistic_fav_wishlist ()	Ermittelt den nach Bewertungen / Aufrufen beliebtesten Wunschzettel.	$number = ""
shop_statistic_fav_wishlist_item ()	Ermittelt den beliebtesten Artikel in den Wunschzetteln.	$number = ""
shop_statistic_max_amount ()	Ermittelt die Kunden mit dem höchsten Umsatz.	$type = "", $form = array()
shop_statistic_max_wishlist ()	Ermittelt den umfangreichsten Wunschzettel.	$number = ""
shop_statistic_report()	Zeigt einen Statistik-Report samt den gewünschten Auswertungen an.	$form
shop_statistic_report_details ()	Ruft eine Funktion einer einzelnen Statistik aus einem Report auf.	$form

shop_statistic_visualize_data ()	Zeigt die Daten der Statistik an.	$array = array()

6.9.287. Datei "/modules/shop/admin/statistic_graph.php"

Funktionen in der Datei "/modules/shop/admin/statistic_graph.php" (Name der Funktion, Beschreibung, Parameter):

shop_statistic_graph()	Generiert eine Grafik, die die Anzahl der Bestellungen im Zeitverlauf anzeigt.	$type, $form
shop_statistic_num_orders ()	Erfasst die Anzahl der Bestellungen in einem bestimmten Zeitraum.	$time_start, $time_stop = ""

6.9.288. Datei "/modules/shop/admin/variants.php"

Funktionen in der Datei "/modules/shop/admin/variants.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_variant()	Zeigt das Formular zum Hinzufügen einer neuen Artikelvariante an.	$item
shop_add_variant_now ()	Fügt eine neue Artikelvariante in die Datenbank ein.	$form, $itemimage
shop_delete_variant()	Löscht eine vorhandene Artikelvariante aus der Datenbank.	$number
shop_edit_variant()	Zeigt das Formular zum Bearbeiten einer vorhandenen Artikelvariante an.	$item, $number
shop_edit_variant_now ()	Aktualisiert die Angaben zu einer vorhandenen Artikelvariante in der Datenbank.	$form, $itemimage
shop_form_variant()	Formular für das Hinzufügen / Bearbeiten von Artikelvarianten.	$item, $number = ""

6.9.289. Datei "/modules/shop/admin/wishlists.php"

Funktionen in der Datei "/modules/shop/admin/wishlists.php" (Name der Funktion, Beschreibung, Parameter):

shop_clean_wishlists()	Entfernt alle Artikel aus den ausgewählten Wunschzetteln.	$array
shop_delete_wishlist()	Entfernt alle Artikel aus dem ausgewählten Wunschzettel samt des Wunschzettels.	$number
shop_delete_wishlists()	Entfernt alle Artikel aus den ausgewählten Wunschzetteln samt der Wunschzettel.	$array
shop_edit_wishlist()	Zeigt das Formular zur Bearbeitung / Überprüfung eines Wunschzettels an.	$number
shop_edit_wishlist_2()	Aktualisiert die Angaben zu einem Wunschzettel eines Kunden.	$number, $form
shop_show_wishlist()	Zeigt die Artikel auf einem Wunschzettel innerhalb eines Iframes an.	$number

6.9.290. Datei "/modules/shop/admin/statistic/articles_cart.php"

Funktionen in der Datei "/modules/shop/admin/statistic/articles_cart.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.291. Datei "/modules/shop/admin/statistic/articles_company.php"

Funktionen in der Datei "/modules/shop/admin/statistic/articles_company.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.292. Datei "/modules/shop/admin/statistic/articles_company_full.php"

Funktionen in der Datei "/modules/shop/admin/statistic/articles_company_full.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.293. Datei "/modules/shop/admin/statistic/articles_salesrank.php"

Funktionen in der Datei "/modules/shop/admin/statistic/articles_salesrank.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.294. Datei "/modules/shop/admin/statistic/articles_views.php"

Funktionen in der Datei "/modules/shop/admin/statistic/articles_views.php" (Name der Funktion, Beschreibung,

Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.295. Datei "/modules/shop/admin/statistic/articles_volume.php"

Funktionen in der Datei "/modules/shop/admin/statistic/articles_volume.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.296. Datei "/modules/shop/admin/statistic/clients_maxamount.php"

Funktionen in der Datei "/modules/shop/admin/statistic/clients_maxamount.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.297. Datei "/modules/shop/admin/statistic/clients_numorders.php"

Funktionen in der Datei "/modules/shop/admin/statistic/clients_numorders.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.298. Datei "/modules/shop/admin/statistic/clients_orders.php"

Funktionen in der Datei "/modules/shop/admin/statistic/clients_orders.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.299. Datei "/modules/shop/admin/statistic/clients_postcode.php"

Funktionen in der Datei "/modules/shop/admin/statistic/clients_postcode.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.300. Datei "/modules/shop/admin/statistic/fav_payment.php"

Funktionen in der Datei "/modules/shop/admin/statistic/fav_payment.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.301. Datei "/modules/shop/admin/statistic/fav_shipment.php"

Funktionen in der Datei "/modules/shop/admin/statistic/fav_shipment.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.302. Datei "/modules/shop/admin/statistic/geo_maps.php"

Funktionen in der Datei "/modules/shop/admin/statistic/geo_maps.php" (Name der Funktion, Beschreibung, Parameter):

grab()	Liest die GEO-Positionen für die Kunden / Bestellungen ein.	$form

kml()	Exportiert die Daten als KML-Datei für Google Earth.	$form
map()	Zeigt die Google Maps-Karte mit den Kunden / Bestellungen an.	$form
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
set_options()	Speichert die Einstellungen zu dieser Statistik ab.	$form
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.303. Datei "/modules/shop/admin/statistic/orders_conversionrate.php"

Funktionen in der Datei "/modules/shop/admin/statistic/orders_conversionrate.php" (Name der Funktion, Beschreibung, Parameter):

calc_week_start()	Ermittelt den Timestamp für den Anfang des Montags in einer bestimmten Woche in einem Jahr.	$week, $year
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.304. Datei "/modules/shop/admin/statistic/orders_graphic.php"

Funktionen in der Datei "/modules/shop/admin/statistic/orders_graphic.php" (Name der Funktion, Beschreibung, Parameter):

num_orders()	Erfasst die Anzahl der Bestellungen in einem bestimmten Zeitraum.	$time_start, $time_stop = "", $array = array()
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.305. Datei "/modules/shop/admin/statistic/orders_perhour.php"

Funktionen in der Datei "/modules/shop/admin/statistic/orders_perhour.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.306. Datei "/modules/shop/admin/statistic/orders_value.php"

Funktionen in der Datei "/modules/shop/admin/statistic/orders_value.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.307. Datei "/modules/shop/admin/statistic/partner_maxamount.php"

Funktionen in der Datei "/modules/shop/admin/statistic/partner_maxamount.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.308. Datei "/modules/shop/admin/statistic/partner_orders.php"

Funktionen in der Datei "/modules/shop/admin/statistic/partner_orders.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.309. Datei "/modules/shop/admin/statistic/revenue_daily.php"

Funktionen in der Datei "/modules/shop/admin/statistic/revenue_daily.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.310. Datei "/modules/shop/admin/statistic/revenue_hours.php"

Funktionen in der Datei "/modules/shop/admin/statistic/revenue_hours.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.311. Datei "/modules/shop/admin/statistic/revenue_monthly.php"

Funktionen in der Datei "/modules/shop/admin/statistic/revenue_monthly.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.312. Datei "/modules/shop/admin/statistic/revenue_quarters.php"

Funktionen in der Datei "/modules/shop/admin/statistic/revenue_quarters.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.313. Datei "/modules/shop/admin/statistic/revenue_weekdays.php"

Funktionen in der Datei "/modules/shop/admin/statistic/revenue_weekdays.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.314. Datei "/modules/shop/admin/statistic/revenue_weekly.php"

Funktionen in der Datei "/modules/shop/admin/statistic/revenue_weekly.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.315. Datei "/modules/shop/admin/statistic/revenue_yearly.php"

Funktionen in der Datei "/modules/shop/admin/statistic/revenue_yearly.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.316. Datei "/modules/shop/admin/statistic/searchword.php"

Funktionen in der Datei "/modules/shop/admin/statistic/searchword.php" (Name der Funktion, Beschreibung, Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.317. Datei "/modules/shop/admin/statistic/section_views.php"

Funktionen in der Datei "/modules/shop/admin/statistic/section_views.php" (Name der Funktion, Beschreibung,

Parameter):

options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.318. Datei "/modules/shop/admin/statistic/template.php"

Funktionen in der Datei "/modules/shop/admin/statistic/template.php" (Name der Funktion, Beschreibung, Parameter):

data()	Ermittelt die Daten der Statistik und liefert diese als Array zurück.	$array = array()
export()	Exportiert die Daten als CSV-Datei.	$array
options()	Grundlegende Einstellungen zu dieser Statistik.	$type = ""
show()	Zeigt die Daten der Statistik an.	$array = array()

6.9.319. Datei "/modules/shop/container/add_box.php"

Funktionen in der Datei "/modules/shop/container/add_box.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_add_box ()	Container für die Anzeige des Formulares zum schnellen Hinzufügen von Artikeln zum Warenkorb.	$form
shop_container_add_box_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.320. Datei "/modules/shop/container/best_reviews.php"

Funktionen in der Datei "/modules/shop/container/best_reviews.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_best_reviews ()	Container für die Anzeige der am besten bewerteten Artikel im Shop.	$form
shop_container_best_reviews_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.321. Datei "/modules/shop/container/bestseller.php"

Funktionen in der Datei "/modules/shop/container/bestseller.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_bestseller ()	Container für die Anzeige der am häufigsten im Shop verkauften Artikel.	$form
shop_container_bestseller_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.322. Datei "/modules/shop/container/breadcrumb_navigation.php"

Funktionen in der Datei "/modules/shop/container/breadcrumb_navigation.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_breadcrumb_navigation()	Container für die Darstellung der "Breadcrumb-/Brotkrümel"-Navigation von der Startseite des Shops zur aktuellen Seite.	$form = array ()
shop_container_breadcrumb_navigation_config()	Liefert Informationen über den Aufbau des Containers zurück.	
shop_container_breadcrumb_navigation_pages()	Liefert das Array mit den verfügbaren Seiten für die Breadcrumb-Navigation zurück (exkl. Kategorien, Artikel und Varianten).	
shop_container_breadcrumb_navigation_sections()	Erstellt die Breadcrumb-Navigation für die Kategorien.	$number, $array

6.9.323. Datei "/modules/shop/container/cart.php"

Funktionen in der Datei "/modules/shop/container/cart.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_cart()	Container für die Anzeige des Warenkorbinhaltes im Shop.	$form
shop_container_cart_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.324. Datei "/modules/shop/container/cart_numtotal.php"

Funktionen in der Datei "/modules/shop/container/cart_numtotal.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_cart_numtotal ()	Container für die Anzahl der momentan im Warenkorb befindlichen Artikel.	$form
shop_container_cart_numtotal_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.325. Datei "/modules/shop/container/cart_numunique.php"

Funktionen in der Datei "/modules/shop/container/cart_numunique.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_cart_numunique ()	Container für die Anzahl der momentan im Warenkorb befindlichen unterschiedlichen Artikel.	$form
shop_container_cart_numunique_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.326. Datei "/modules/shop/container/cart_price.php"

Funktionen in der Datei "/modules/shop/container/cart_price.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_cart_price ()	Container für die Anzeige des Gesamtpreises (brutto).	$form
shop_container_cart_price_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.327. Datei "/modules/shop/container/categories.php"

Funktionen in der Datei "/modules/shop/container/categories.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_categories ()	Container für die Anzeige von Kategorien / der Kategoriestruktur.	$form
shop_container_categories_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.328. Datei "/modules/shop/container/category_box.php"

Funktionen in der Datei "/modules/shop/container/category_box.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_category_box ()	Container für das Auswahlmenü für Kategorien.	$form
shop_container_category_box_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
shop_section_box()	Generiert die Liste der Kategorien für das Auswahlmenü.	$level, $count = "0"

6.9.329. Datei "/modules/shop/container/category_list.php"

Funktionen in der Datei "/modules/shop/container/category_list.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_category_list ()	Container für das Anzeige der Kategorien, in welchen ein Artikel enthalten ist.	$form
shop_container_category_list_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
shop_container_category_list_path ()	Generiert den Pfad der Kategorien eines Artikels, von der untersten bis zur obersten Ebene.	$number

6.9.330. Datei "/modules/shop/container/companies.php"

Funktionen in der Datei "/modules/shop/container/companies.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_companies ()	Container für die Anzeige von Herstellerlisten und -details.	$form
shop_container_companies_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.331. Datei "/modules/shop/container/comparison.php"

Funktionen in der Datei "/modules/shop/container/comparison.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_comparison ()	Container für die Anzeige der vom Kunden zum Vergleich ausgewählten Artikel.	$form
shop_container_comparison_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.332. Datei "/modules/shop/container/crossselling.php"

Funktionen in der Datei "/modules/shop/container/crossselling.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_crossselling	Container für die Anzeige von Cross-Selling-Artikeln / Zubehör / Bundles etc.	$form

()		
shop_container_crossselling_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.333. Datei "/modules/shop/container/currencies.php"

Funktionen in der Datei "/modules/shop/container/currencies.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_currencies ()	Container für die Anzeige der Artikelpreise in verschiedenen Währungen.	$form
shop_container_currencies_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.334. Datei "/modules/shop/container/currency_box.php"

Funktionen in der Datei "/modules/shop/container/currency_box.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_currency_box ()	Container für die Auswahl der Währungen.	$form
shop_container_currency_box_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.335. Datei "/modules/shop/container/custom_navigation.php"

Funktionen in der Datei "/modules/shop/container/custom_navigation.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_custom_navigation ()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Dies sind u.a. die verschiedenen vorgefertigten Navigationstypen. Aufruf in den Vorlagen: {cp:container module="shop" type="custom_navigation" category="0" template="modules/shop/templates/container/sections.html" template_selected="modules/shop/templates/container/sections_selected.html"}	$form
shop_container_custom_navigation_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
shop_navigation_get_root ()	Ermittelt den Hauptnavigationspunkt für die Anzeige der entsprechenden Vorlage.	$section = "", $array = array()

6.9.336. Datei "/modules/shop/container/delivery.php"

Funktionen in der Datei "/modules/shop/container/delivery.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_delivery ()	Container für die Darstellung der möglichen Lieferstati.	$form
shop_container_delivery_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.337. Datei "/modules/shop/container/details.php"

Funktionen in der Datei "/modules/shop/container/details.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_details()	Container für die Darstellung eines einzelnen Artikels.	$form
shop_container_details_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.338. Datei "/modules/shop/container/ekomi.php"

Funktionen in der Datei "/modules/shop/container/ekomi.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_ekomi()	Container für die Anzeige der bei eKomi abgegebenen Bewertungen der Kunden.	$form
shop_container_ekomi_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.339. Datei "/modules/shop/container/filter_sections.php"

Funktionen in der Datei "/modules/shop/container/filter_sections.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_filter_sections ()	Container für die Anzeige des Formulars zur Filterung der Artikel innerhalb einer Kategorie.	$form
shop_container_filter_sections_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.340. Datei "/modules/shop/container/interactive_attributes.php"

Funktionen in der Datei "/modules/shop/container/interactive_attributes.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_interactive_attributes()	Container für die Anzeige der Interaktiven Attribute zu einem Artikel.	$form
shop_container_interactive_attributes_config()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.341. Datei "/modules/shop/container/last_articles.php"

Funktionen in der Datei "/modules/shop/container/last_articles.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_last_articles ()	Container für die Anzeige der zuletzt im Shop aufgenommenen Artikel.	$form
shop_container_last_articles_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.342. Datei "/modules/shop/container/last_order.php"

Funktionen in der Datei "/modules/shop/container/last_order.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_last_order ()	Container für die Anzeige der zuletzt gekauften Artikel.	$form
shop_container_last_order_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.343. Datei "/modules/shop/container/lastvisited.php"

Funktionen in der Datei "/modules/shop/container/lastvisited.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_lastvisited ()	Container für die Anzeige der zuletzt aufgerufenen Artikel.	$form
shop_container_lastvisited_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.344. Datei "/modules/shop/container/list.php"

Funktionen in der Datei "/modules/shop/container/list.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_list()	Container für die Anzeige von beliebigen Artikelauflistungen.	$form
shop_container_list_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.345. Datei "/modules/shop/container/login_box.php"

Funktionen in der Datei "/modules/shop/container/login_box.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_login_box ()	Container für die Anzeige des Login-Formulares bzw. der Kundenoptionen.	$form
shop_container_login_box_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.346. Datei "/modules/shop/container/media.php"

Funktionen in der Datei "/modules/shop/container/media.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_media()	Container für die Anzeige der Mediendateien für einen Artikel.	$form
shop_container_media_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.347. Datei "/modules/shop/container/owner_ratings.php"

Funktionen in der Datei "/modules/shop/container/owner_ratings.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_owner_rating_types ()	Liefert die verfügbaren Typen für die Bewertung der Artikel zurück.	
shop_container_owner_ratings ()	Container für die Anzeige der Bewertung eines Artikels durch den Shop-Betreiber.	$form
shop_container_owner_ratings_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.348. Datei "/modules/shop/container/payment.php"

Funktionen in der Datei "/modules/shop/container/payment.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_payment ()	Container für die Anzeige einer detaillierten Beschreibung der Zahlungsarten.	$form
shop_container_payment_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.349. Datei "/modules/shop/container/prices.php"

Funktionen in der Datei "/modules/shop/container/prices.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_prices()	Container für die Anzeige der Staffelpreise zu einem Artikel.	$form
shop_container_prices_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.350. Datei "/modules/shop/container/pricetrend.php"

Funktionen in der Datei "/modules/shop/container/pricetrend.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_pricetrend ()	Container für die Anzeige eines Preistrends.	$form
shop_container_pricetrend_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.351. Datei "/modules/shop/container/random.php"

Funktionen in der Datei "/modules/shop/container/random.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_random ()	Container für die Anzeige von zufällig ausgewählten Artikeln.	$form
shop_container_random_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.352. Datei "/modules/shop/container/random_coupons.php"

Funktionen in der Datei "/modules/shop/container/random_coupons.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_random_coupons ()	Container für die zufällige Anzeige von Gutscheincodes an beliebigen Stellen der Website. Aufruf in den Vorlagen: {cp:container module="shop" type="random_coupons" frequency="20"}	$forn
shop_container_random_coupons_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.353. Datei "/modules/shop/container/related.php"

Funktionen in der Datei "/modules/shop/container/related.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_related ()	Container für die Anzeige von Artikeln à la "Kunden, die x gekauft haben...".	$form
shop_container_related_config ()	Liefert Informationen über den Aufbau des Containers zurück.	
shop_get_related()	Ermittelt die verwandten Artikel für "Kunden, die x gekauft haben...".	$number

6.9.354. Datei "/modules/shop/container/review_box.php"

Funktionen in der Datei "/modules/shop/container/review_box.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_review_box ()	Container für die Anzeige des Formulares für Kundenmeinungen.	$form
shop_container_review_box_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.355. Datei "/modules/shop/container/reviews.php"

Funktionen in der Datei "/modules/shop/container/reviews.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_reviews ()	Container für die Anzeige von Kundenmeinungen im Shop.	$form
shop_container_reviews_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.356. Datei "/modules/shop/container/search_box.php"

Funktionen in der Datei "/modules/shop/container/search_box.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_search_box ()	Container für die Anzeige des Formulares zur Suche im Shop.	$form
shop_container_search_box_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.357. Datei "/modules/shop/container/searches.php"

Funktionen in der Datei "/modules/shop/container/searches.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_searches ()	Container für die Anzeige der im Shop durchgeführten Suchanfragen.	$form
shop_container_searches_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.358. Datei "/modules/shop/container/shipment.php"

Funktionen in der Datei "/modules/shop/container/shipment.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_shipment ()	Container für die Anzeige einer detaillierten Beschreibung der Versandarten.	$form
shop_container_shipment_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.359. Datei "/modules/shop/container/tag_cloud.php"

Funktionen in der Datei "/modules/shop/container/tag_cloud.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_tag_cloud ()	Container für die Anzeige der wichtigsten Keywords im Shop als Tagcloud. Aufruf in den Vorlagen: {cp:container module="xservices" type="tag_cloud" limit="..."}	$forn
shop_container_tag_cloud_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.360. Datei "/modules/shop/container/variants.php"

Funktionen in der Datei "/modules/shop/container/variants.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_variants ()	Container für die Anzeige von Artikelvarianten.	$form
shop_container_variants_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.361. Datei "/modules/shop/container/wishlist.php"

Funktionen in der Datei "/modules/shop/container/wishlist.php" (Name der Funktion, Beschreibung, Parameter):

shop_container_wishlist ()	Container für die Anzeige der Artikel eines öffentlich verfügbaren Wunschzettels.	$form
shop_container_wishlist_config ()	Liefert Informationen über den Aufbau des Containers zurück.	

6.9.362. Datei "/modules/shop/export/articles/billiger.de.php"

Funktionen in der Datei "/modules/shop/export/articles/billiger.de.php" (Name der Funktion, Beschreibung, Parameter):

shop_billiger_sections()	Erstellt die Breadcrumb-Navigation für die Kategorien.	$number
shop_export_option()	Erzeugt den Export der Artikel für billiger.de. Weitere Informationen: http://www.billiger.de/application/, http://www.billiger.de/application/tech.html CSV-Testdatei für Datensätze: http://www.billiger.de/application/documents/vorlage.csv	$form = array()

6.9.363. Datei "/modules/shop/export/articles/ciao.php"

Funktionen in der Datei "/modules/shop/export/articles/ciao.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Ciao.de.	$form = array()

6.9.364. Datei "/modules/shop/export/articles/evendi.php"

Funktionen in der Datei "/modules/shop/export/articles/evendi.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Evendi.	$form = array()

6.9.365. Datei "/modules/shop/export/articles/froogle.php"

Funktionen in der Datei "/modules/shop/export/articles/froogle.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Froogle.	$form = array()
shop_froogle_escape()	Reinigt die Inhalte der Formularfelder für Froogle.	$string
shop_froogle_navigation ()	Erzeugt die Breadcrumb-Navigation für Froogle.	$number

6.9.366. Datei "/modules/shop/export/articles/geizhals.at.php"

Funktionen in der Datei "/modules/shop/export/articles/geizhals.at.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Geizhals.at.	$form = array()

6.9.367. Datei "/modules/shop/export/articles/getprice.php"

Funktionen in der Datei "/modules/shop/export/articles/getprice.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Getprice.	$form = array()
shop_getprice_escape()	Reinigt die Inhalte der Formularfelder für Geizhals.at.	$string

6.9.368. Datei "/modules/shop/export/articles/google_base.php"

Funktionen in der Datei "/modules/shop/export/articles/google_base.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Google Base (http://base.google.de/). Anleitung: http://base.google.com/base/de/tab_instructions.html	$form = array()

6.9.369. Datei "/modules/shop/export/articles/hitmeister.php"

Funktionen in der Datei "/modules/shop/export/articles/hitmeister.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Hitmeister.de, Importformat Version 1.2 vom Dezember 2007. Zusätzlich in der Datenbanktabelle *_shop_items erforderliche Felder für Hitmeister: custom_hitmeister_condition, custom_hitmeister_comment	$form = array()
shop_hitmeister_escape ()	Reinigt die Inhalte der Formularfelder für Geizhals.at.	$string

6.9.370. Datei "/modules/shop/export/articles/html.php"

Funktionen in der Datei "/modules/shop/export/articles/html.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel im HTML-Format.	$form = array()

6.9.371. Datei "/modules/shop/export/articles/idealo.php"

Funktionen in der Datei "/modules/shop/export/articles/idealo.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Google Base (http://base.google.de/). Anleitung: http://base.google.com/base/de/tab_instructions.html	$form = array()
shop_export_option_section_path ()	Erstellt die Pfad-Navigation für die Kategorien.	$number

6.9.372. Datei "/modules/shop/export/articles/kelkoo.php"

Funktionen in der Datei "/modules/shop/export/articles/kelkoo.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Kelkoo.	$form = array()

6.9.373. Datei "/modules/shop/export/articles/kelkoo_premium.php"

Funktionen in der Datei "/modules/shop/export/articles/kelkoo_premium.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Kelkoo Premium.	$form = array()

6.9.374. Datei "/modules/shop/export/articles/milando.php"

Funktionen in der Datei "/modules/shop/export/articles/milando.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Milando.	$form = array()
shop_milando_escape()	Reinigt die Inhalte der Formularfelder für Milando.	$string

6.9.375. Datei "/modules/shop/export/articles/pangora.php"

Funktionen in der Datei "/modules/shop/export/articles/pangora.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Pangora.	$form = array()

6.9.376. Datei "/modules/shop/export/articles/preisauskunft.de.php"

Funktionen in der Datei "/modules/shop/export/articles/preisauskunft.de.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Preisauskunft.de.	$form = array()

6.9.377. Datei "/modules/shop/export/articles/preisroboter.de.php"

Funktionen in der Datei "/modules/shop/export/articles/preisroboter.de.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Preisroboter.de gemäß "preisroboter.txt V2.0" (http://www.preisroboter.de/preisroboter_info_anbieterinfos.php#preisroboter_txt2).	$form = array()

6.9.378. Datei "/modules/shop/export/articles/preissuchmaschine.de.php"

Funktionen in der Datei "/modules/shop/export/articles/preissuchmaschine.de.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Preissuchmaschine.de.	$form = array()

6.9.379. Datei "/modules/shop/export/articles/preiswal.php"

Funktionen in der Datei "/modules/shop/export/articles/preiswal.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Preiswal.de.	$form = array()
shop_preiswal_escape()	Reinigt die Inhalte der Formularfelder für Preiswal.de.	$string

6.9.380. Datei "/modules/shop/export/articles/schnaeppchenjagd.de.php"

Funktionen in der Datei "/modules/shop/export/articles/schnaeppchenjagd.de.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Schnaeppchenjagd.de.	$form = array()

6.9.381. Datei "/modules/shop/export/articles/schottenland.php"

Funktionen in der Datei "/modules/shop/export/articles/schottenland.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Schottenland.	$form = array()
shop_schottenland_escape ()	Reinigt die Inhalte der Formularfelder für Schottenland.	$string

6.9.382. Datei "/modules/shop/export/articles/shopinfo.xml-items.php"

Funktionen in der Datei "/modules/shop/export/articles/shopinfo.xml-items.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel im shopinfo.xml-Standard (Produktdatei).	$form = array()
shop_shopinfo_escape()	Reinigt die Inhalte der Formularfelder für den Export.	$string

6.9.383. Datei "/modules/shop/export/articles/shopinfo.xml.php"

Funktionen in der Datei "/modules/shop/export/articles/shopinfo.xml.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel im shopinfo.xml-Standard (Shopdatei).	$form = array()
shopinfo_options()	Liefert ein Array mit Einstellungen für die Erzeugung der "shopinfo.xml"-Datei zurück.	
shopinfo_urls()	Konvertiert die URL's in der shopinfo.xml-Datei.	$url, $name

6.9.384. Datei "/modules/shop/export/articles/shopzilla.php"

Funktionen in der Datei "/modules/shop/export/articles/shopzilla.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für shopzilla. Upload/Infos unter http://merchant.shopzilla.de/.	$form = array()
shop_shopzilla_escape()	Reinigt die Inhalte der Formularfelder für Shopzilla.	$string

6.9.385. Datei "/modules/shop/export/articles/wir-lieben-preise.php"

Funktionen in der Datei "/modules/shop/export/articles/wir-lieben-preise.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Wir-Lieben-Preise.de. Weitere Informationen: http://wir-lieben-preise.de/export-spezifikation.php, http://www.wir-lieben-preise.de/downloads.php CSV-Testdatei für Datensätze: http://wir-lieben-preise.de/exporter_dl/wirliebenpreise.txt	$form = array()

6.9.386. Datei "/modules/shop/export/articles/yatego.php"

Funktionen in der Datei "/modules/shop/export/articles/yatego.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Yatego.	$form = array()
shop_yatego_escape()	Reinigt die Inhalte der Formularfelder für Preiscafe.	$string
shop_yatego_sections()	Liefert die Zuordnung für die Yatego-Kategorien zurück. Hier müssen Sie die Kategorie-ID's im CP::Shop den Yatego-Kategorien zuordnen. Z.B. "13" => "17-21-02". Auch mehre Yatego-Kategorien sind möglich. Diese sind dann durch "," zu trennen.	

6.9.387. Datei "/modules/shop/export/clients/google_mail.php"

Funktionen in der Datei "/modules/shop/export/clients/google_mail.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Artikel für Google Mail (http://mail.google.com/). Anleitung: http://mail.google.com/support/bin/answer.py?answer=12119, http://mail.google.com/support/bin/answer.py?answer=14024	$form = array()

6.9.388. Datei "/modules/shop/export/orders/AFS-Kaufmann.php"

Funktionen in der Datei "/modules/shop/export/orders/AFS-Kaufmann.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Bestellungen für AFS Kaufmann.	$form = array()

6.9.389. Datei "/modules/shop/export/orders/SQL-Export.php"

Funktionen in der Datei "/modules/shop/export/orders/SQL-Export.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Bestellungen im SQL-Format.	$form = array()

6.9.390. Datei "/modules/shop/export/orders/inkasso.MediaFinanz.php"

Funktionen in der Datei "/modules/shop/export/orders/inkasso.MediaFinanz.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_escape()	Reinigt die Inhalte der Formularfelder für den Export.	$string
shop_export_option()	Erzeugt den Export der Bestellungen für die mediafinanz AG.	$form = array()

6.9.391. Datei "/modules/shop/export/orders/lexware.php"

Funktionen in der Datei "/modules/shop/export/orders/lexware.php" (Name der Funktion, Beschreibung, Parameter):

shop_export_option()	Erzeugt den Export der Bestellungen für Lexware.	$form = array()
shop_lexware_escape()	Reinigt die Inhalte der Formularfelder für Lexware.	$string

6.9.392. Datei "/modules/shop/import/affilinet.php"

Funktionen in der Datei "/modules/shop/import/affilinet.php" (Name der Funktion, Beschreibung, Parameter):

shop_import_extern()	Importiert Artikel aus den CSV-Dateien von Affili.net. -> 1. Zeile im Format: "ArtNumber";"Title";"Description_Short";"DisplayPrice";"Img_Url";"Deeplink1";ProductCategoryID;"ProductCategoryName" -> Unterstützt werden weiterhin die zusätzlichen Felder "Description", "Keywords", "Brand", "Distributor", "EAN" -> Trennzeichen: ";", Umgeben von: """	$file_nam

6.9.393. Datei "/modules/shop/import/ciando-csv.php"

Funktionen in der Datei "/modules/shop/import/ciando-csv.php" (Name der Funktion, Beschreibung, Parameter):

shop_import_extern()	Importiert Artikeldaten im CSV-Format von www.ciando.com. Aufbau: Autor;Deeplink;Bildlink;ISBN;Titel;Jahr;PREIS;Verlag;Kurzinfo	$file_nam

6.9.394. Datei "/modules/shop/import/mgshopx.php"

Funktionen in der Datei "/modules/shop/import/mgshopx.php" (Name der Funktion, Beschreibung, Parameter):

shop_import_extern()	-> Im Ordner "modules/shop/import/" müssen die Dateien "artikeldaten.csv", "artikelbilder.csv" und "kategorien.csv" vorhanden sein. -> Bilder sind im Ordner "modules/shop/images/custom/" abzulegen. ACHTUNG: Der Import über diese Importdefinition löscht zuerst alle vorhandenen Artikel, Hersteller, Mediendateien und Kategorien!	$file_nam

6.9.395. Datei "/modules/shop/information/schnaeppchenjagd.de.php"

Funktionen in der Datei "/modules/shop/information/schnaeppchenjagd.de.php" (Name der Funktion, Beschreibung, Parameter):

shop_extern_show()	Zeigt im Administrationsbereich Informationen zu Schnaeppchenjagd.de an.	

6.9.396. Datei "/modules/shop/payment/clickandbuy.php"

Funktionen in der Datei "/modules/shop/payment/clickandbuy.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter. URL für die Speicherung im Servicebereich von ClickandBuy: -> http://www.[DOMAINNAME].de/[PFAD]/index.php?file=shop&mode=gateway_extern&type=clickandbuy&show=confirmation&clean=yes	$form

6.9.397. Datei "/modules/shop/payment/directebanking.php"

Funktionen in der Datei "/modules/shop/payment/directebanking.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.398. Datei "/modules/shop/payment/expercash.php"

Funktionen in der Datei "/modules/shop/payment/expercash.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
helper_parse_xml()	Von ExperCash zurückgegebene Daten im XML-Format werden zerlegt.	$xml, $pattern

information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.399. Datei "/modules/shop/payment/expercash_iframe.php"

Funktionen in der Datei "/modules/shop/payment/expercash_iframe.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.400. Datei "/modules/shop/payment/gzspaymaster.php"

Funktionen in der Datei "/modules/shop/payment/gzspaymaster.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.401. Datei "/modules/shop/payment/infin.php"

Funktionen in der Datei "/modules/shop/payment/infin.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter. URL für die Rückgabe des Zahlungsergebnissen an den CP::Shop durch infin: -> http://www.[DOMAINNAME].de/[PFAD]/index.php?file=shop&mode=gateway_extern&type=infin&show=confirmation&clean=yes	$form

6.9.402. Datei "/modules/shop/payment/mpay24.php"

Funktionen in der Datei "/modules/shop/payment/mpay24.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array(
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array(
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter. URL's für die Speicherung im MerchantProfile: -> http://www.[DOMAINNAME].de/[PFAD]/index.php?file=shop&mode=gateway_extern&type=mpay24&show=confirmation&clean=yes -> http://www.[DOMAINNAME].de/[PFAD]/index.php?file=shop&mode=gateway_extern&type=mpay24&show=success -> http://www.[DOMAINNAME].de/[PFAD]/index.php?file=shop&mode=gateway_extern&type=mpay24&show=error	$form

6.9.403. Datei "/modules/shop/payment/paypal.php"

Funktionen in der Datei "/modules/shop/payment/paypal.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array()
helper_call()	Interne Methode dieser Zahlungsart. Greift per CURL-Erweiterung auf den PayPal-Server zu.	$method, $string
helper_format_string()	Interne Methode dieser Zahlungsart. Wandelt einen String in ein Assoziatives Array um.	$string
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor. Dokumentation der PayPal Starter-Schnittstelle: http://www.paypal.com/de_DE/html/IntegrationCenter/ic_pps_home.html	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.404. Datei "/modules/shop/payment/paysolution.php"

Funktionen in der Datei "/modules/shop/payment/paysolution.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
hmacsha1()	Berechnet den verschlüsselten MAC-Code für die Übergabe der Zahlungsdaten.	$key, $data
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.405. Datei "/modules/shop/payment/saferpay.php"

Funktionen in der Datei "/modules/shop/payment/saferpay.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array(
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array(
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter. URL's für die Speicherung im Account: -> http://www.[DOMAINNAME].de/[PFAD]/index.php?file=shop&mode=gateway_extern&type=saferpay&show=confirmation&clean=yes -> http://www.[DOMAINNAME].de/[PFAD]/index.php?file=shop&mode=gateway_extern&type=saferpay&show=success -> http://www.[DOMAINNAME].de/[PFAD]/index.php?file=shop&mode=gateway_extern&type=saferpay&show=error	$form

6.9.406. Datei "/modules/shop/payment/sofortueberweisung.php"

Funktionen in der Datei "/modules/shop/payment/sofortueberweisung.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.407. Datei "/modules/shop/payment/template.php"

Funktionen in der Datei "/modules/shop/payment/template.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array

		()
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.408. Datei "/modules/shop/payment/tpay.php"

Funktionen in der Datei "/modules/shop/payment/tpay.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.409. Datei "/modules/shop/payment/worldpay.php"

Funktionen in der Datei "/modules/shop/payment/worldpay.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.410. Datei "/modules/shop/payment/yellowpay.php"

Funktionen in der Datei "/modules/shop/payment/yellowpay.php" (Name der Funktion, Beschreibung, Parameter):

create()	Erzeugt das erweiterte Formular für den Administrationsbereich.	$form = array ()
form()	Zeigt weitere Eingabefelder innerhalb der Übersicht der Zahlungsarten im Bestellprozess an.	$form = array ()
information()	Zeigt weitere Informationen zu dieser Zahlungsart unterhalb des Formulares an (z.B. eine Übersicht der Transaktionen).	
options()	Grundlegende Einstellungen zu dieser Zahlungsart.	$type = ""
process()	Verarbeitet die Daten und bereitet die Übergabe vor.	$form
receive()	Erhält die zurückgelieferten Daten vom Zahlungsanbieter.	$form

6.9.411. Datei "/modules/shop/public/account.php"

Funktionen in der Datei "/modules/shop/public/account.php" (Name der Funktion, Beschreibung, Parameter):

shop_account()	Zeigt das Kunden-Konto mit den letzten Bestellungen für einen eingeloggten Kunden an.	$form

6.9.412. Datei "/modules/shop/public/buy.php"

Funktionen in der Datei "/modules/shop/public/buy.php" (Name der Funktion, Beschreibung, Parameter):

shop_buy()	Führt eine Bestellung komplett durch mit allen dazugehörigen Schritten (Emailversand, Rechnungserstellung usw.).	
shop_buy_error()	Gibt eine Meldung nach einer gescheiterten Bestellung aus.	$type
shop_buy_success()	Gibt eine Meldung nach einer erfolgreichen Bestellung aus.	

6.9.413. Datei "/modules/shop/public/cart_0.php"

Funktionen in der Datei "/modules/shop/public/cart_0.php" (Name der Funktion, Beschreibung, Parameter):

shop_cart()	Erster Schritt im Bestellprozess mit der Darstellung des Warenkorbs.	

6.9.414. Datei "/modules/shop/public/cart_1.php"

Funktionen in der Datei "/modules/shop/public/cart_1.php" (Name der Funktion, Beschreibung, Parameter):

shop_cart_1()	Zweiter Schritt im Bestellprozess. Der Kunde registriert sich / loggt sich ein. Wenn der Kunde bereits eingeloggt sein sollte, dann wird die Express-Bestellung angezeigt, sofern diese aktiviert ist.	$forn

6.9.415. Datei "/modules/shop/public/cart_2.php"

Funktionen in der Datei "/modules/shop/public/cart_2.php" (Name der Funktion, Beschreibung, Parameter):

shop_cart_2()	Dritter Schritt im Bestellprozess. Der Kunde wählt die gewünschten Zahlungs- und Versandarten aus.	$form
shop_cart_2_now()	Nimmt die Daten eines Neukunden entgegen und leitet zum dritten Schritt im Bestellprozess weiter.	$form

6.9.416. Datei "/modules/shop/public/cart_3.php"

Funktionen in der Datei "/modules/shop/public/cart_3.php" (Name der Funktion, Beschreibung, Parameter):

shop_cart_3()	Vierter Schritt im Bestellprozess. Alle eingegebenen Daten werden bestätigt.	$form
shop_cart_3_error()	Gibt einen Hinweis an den Benutzer aus, wenn der Bestellwert zu gering / zu hoch ist.	
shop_cart_3_now()	Nimmt die Daten zu Zahlungs- und Versandart entgegen und leitet zum vierten Schritt im Bestellprozess weiter.	$form

6.9.417. Datei "/modules/shop/public/cart_add.php"

Funktionen in der Datei "/modules/shop/public/cart_add.php" (Name der Funktion, Beschreibung, Parameter):

shop_add()	Fügt einen Artikel zum Warenkorb hinzu unter Beachtung von Lieferzeit, Freigabe, noch vorrätiger Anzahl etc.	$number, $amount, $return = "0"
shop_add_interactive()	Fügt einen Artikel zum Warenkorb hinzu samt den Werten für die Interaktiven Attribute.	$number, $form = array()

6.9.418. Datei "/modules/shop/public/cart_change.php"

Funktionen in der Datei "/modules/shop/public/cart_change.php" (Name der Funktion, Beschreibung, Parameter):

shop_cart_change()	Gleiche Funktion wie shop_change_values() jedoch für einzelne Artikel per "+"/"-"-Buttons.	$action, $number
shop_change_values()	Aktualisiert den Artikelbestand im Warenkorb eines Kunden für mehrere Artikel gleichzeitig.	$form

6.9.419. Datei "/modules/shop/public/cart_empty.php"

Funktionen in der Datei "/modules/shop/public/cart_empty.php" (Name der Funktion, Beschreibung, Parameter):

shop_cart_empty()	Löscht den kompletten Warenkorb eines Besuchers / Kunden.	$form

6.9.420. Datei "/modules/shop/public/cart_sendto.php"

Funktionen in der Datei "/modules/shop/public/cart_sendto.php" (Name der Funktion, Beschreibung, Parameter):

shop_cart_sendto()	Liefert im Bestellprozess die Eingabeseite für die Lieferanschrift.	$form
shop_cart_sendto_address ()	Lädt eine Lieferanschrift eines eingeloggten Kunden für diese Bestellung.	$address

6.9.421. Datei "/modules/shop/public/client_address.php"

Funktionen in der Datei "/modules/shop/public/client_address.php" (Name der Funktion, Beschreibung, Parameter):

shop_client_address()	Zeigt die Lieferadressen eines Kunden im geschlossenen Kunden-Konto an.	$form
shop_client_address_2()	Fügt eine neue Lieferadresse zum Adressbuch eines Kunden hinzu.	$form
shop_client_delete_address ()	Entfernt eine vorhandene Lieferadresse eines Kunden aus der Datenbank.	$number

6.9.422. Datei "/modules/shop/public/client_edit_order.php"

Funktionen in der Datei "/modules/shop/public/client_edit_order.php" (Name der Funktion, Beschreibung, Parameter):

shop_client_edit_order()	Zeigt die Details zu einer Bestellung an inkl. Änderungsmöglichkeiten durch den Kunden.	$number
shop_client_edit_order_2 ()	Aktualisiert die Angaben zu einer Bestellung (z.B. Stornierung von Artikeln).	$form

6.9.423. Datei "/modules/shop/public/client_options.php"

Funktionen in der Datei "/modules/shop/public/client_options.php" (Name der Funktion, Beschreibung, Parameter):

shop_client_options()	Zeigt ein Formular für die Aktualisierung der persönlichen Daten an.	$form
shop_client_options_2()	Führt die Aktualisierung der persönlichen Daten des Kunden im Shop durch.	$form

6.9.424. Datei "/modules/shop/public/client_password.php"

Funktionen in der Datei "/modules/shop/public/client_password.php" (Name der Funktion, Beschreibung, Parameter):

shop_client_password()	Zeigt ein Formular für die Aktualisierung des Passwortes durch den Kunden an.	$form
shop_client_password_2 ()	Aktualisiert die Angaben zum Passwort eines Kunden im Kunden-Bereich.	$form

6.9.425. Datei "/modules/shop/public/client_payment.php"

Funktionen in der Datei "/modules/shop/public/client_payment.php" (Name der Funktion, Beschreibung, Parameter):

shop_client_payment()	Zeigt die Zahlungseinstellungen eines Kunden im geschlossenen Kunden-Konto an.	$form
shop_client_payment_2 ()	Aktualisiert die Angaben zu den Zahlungseinstellungen eines Kunden im Kunden-Bereich.	$form

6.9.426. Datei "/modules/shop/public/client_voucher.php"

Funktionen in der Datei "/modules/shop/public/client_voucher.php" (Name der Funktion, Beschreibung, Parameter):

shop_client_coupons()	Zeigt die Gutscheine eines Kunden im geschlossenen Kunden-Konto an.	$form

6.9.427. Datei "/modules/shop/public/client_wishlist.php"

Funktionen in der Datei "/modules/shop/public/client_wishlist.php" (Name der Funktion, Beschreibung, Parameter):

shop_client_wishlist()	Zeigt den Wunschzettel eines Kunden innerhalb des Kunden-Kontos an. URL: index.php?file=shop&mode=client_wishlist	$form
shop_client_wishlist_add ()	Fügt für einen Kunden einen neuen Wunschzettel zur Datenbank hinzu. URL: index.php?file=shop&mode=client_wishlist_add&form[...]=[...]	$form
shop_client_wishlist_additem ()	Fügt für einen Kunden einen neuen Wunschzettel zur Datenbank hinzu. URL: index.php?file=shop&mode=client_wishlist_additem&number=[ID]	$number
shop_client_wishlist_delete ()	Löscht den / einen Wunschzettel eines Kunden inklusive der darin enthaltenen Artikel. URL: index.php?file=shop&mode=cliet_wishlist_delete[&number=[ID]]	$number = "
shop_client_wishlist_deleteitem ()	Entfernt einen Artikel von einem Wunschzettel eines Kunden. URL: index.php?file=shop&mode=client_wishlist_deleteitem&number=[ID]	$number
shop_client_wishlist_deletelist ()	Entfernt alle Artikel von einem Wunschzettel eines Kunden. URL: index.php?file=shop&mode=client_wishlist_deletelist&number=[ID]	$number
shop_client_wishlist_edit ()	Aktualisiert die Angaben zum Wunschzettel eines Kunden in der Datenbank. URL: index.php?file=shop&mode=client_wishlist_edit&form[...]=[...]	$form
shop_client_wishlist_item_fromcart ()	Transferiert einen Artikel aus dem Warenkorb in einen Wunschzettel eines Kunden. URL: index.php?file=shop&mode=client_wishlist_item_fromcart&number=[ID]	$number
shop_client_wishlist_item_tocart ()	Transferiert einen Artikel aus einem Wunschzettel in den Warenkorb eines Kunden. URL: index.php?file=shop&mode=client_wishlist_item_tocart&number=[ID]	$number
shop_client_wishlist_list_fromcart ()	Transferiert alle Artikel aus dem Warenkorb in einen Wunschzettel eines Kunden. URL: index.php?file=shop&mode=client_wishlist_list_fromcart&number=[ID]	
shop_client_wishlist_list_tocart ()	Transferiert alle Artikel aus einem Wunschzettel in den Warenkorb eines Kunden. URL: index.php?file=shop&mode=client_wishlist_list_tocart&number=[ID]	$number
shop_client_wishlist_move ()	Verschiebt einen Artikel zwischen mehreren Wunschzetteln eines Kunden. URL: index.php?file=shop&mode=client_wishlist_move&number=[ID]&wishlist=[ID]	$number, $wishlist

6.9.428. Datei "/modules/shop/public/comparison.php"

Funktionen in der Datei "/modules/shop/public/comparison.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_comparison()	Fügt einen Artikel zum Artikelvergleich hinzu.	$number, $form = array()
shop_count_comparison ()	Zählt die im Artikelvergleich liegenden Artikel.	
shop_delete_comparison ()	Entfernt einen Artikel aus dem Artikelvergleich eines Kunden.	$number, $form = array()

6.9.429. Datei "/modules/shop/public/coupons.php"

Funktionen in der Datei "/modules/shop/public/coupons.php" (Name der Funktion, Beschreibung, Parameter):

shop_coupon_check()	Leitet die Überprüfung des Gutscheincodes ein und wechselt dann zurück auf die vorherige Seite.	$code, $email = ""
shop_coupon_check_now ()	Überprüft, ob ein Neu- oder Stammkunde einen übergebenen Gutscheincode nutzen kann.	$code, $email = ""

6.9.430. Datei "/modules/shop/public/details.php"

Funktionen in der Datei "/modules/shop/public/details.php" (Name der Funktion, Beschreibung, Parameter):

shop_details()	Zeigt einen einzelnen Artikel in der Detailansicht an.	$number

6.9.431. Datei "/modules/shop/public/download.php"

Funktionen in der Datei "/modules/shop/public/download.php" (Name der Funktion, Beschreibung, Parameter):

shop_download()	Stellt eine Datei eines bestellten Artikels aus einer Bestellung für den eingeloggten Kunden zum Download bereit.	$number

6.9.432. Datei "/modules/shop/public/email_available.php"

Funktionen in der Datei "/modules/shop/public/email_available.php" (Name der Funktion, Beschreibung, Parameter):

shop_email_available()	Verschickt die Information über Artikel mit einem geringen Warenbestand an den Shop-Betreiber.	$number

6.9.433. Datei "/modules/shop/public/email_company.php"

Funktionen in der Datei "/modules/shop/public/email_company.php" (Name der Funktion, Beschreibung, Parameter):

shop_company_mail()	Info-Email an einen Hersteller, sobald einer seiner Artikel ausgegangen ist.	$items, $order = ""

6.9.434. Datei "/modules/shop/public/email_order.php"

Funktionen in der Datei "/modules/shop/public/email_order.php" (Name der Funktion, Beschreibung, Parameter):

shop_order_mail()	Stellt die Bestellbestätigungs-Email für einen Kunden zusammen.	$array, $order, $send_emails = "1"

6.9.435. Datei "/modules/shop/public/email_partner.php"

Funktionen in der Datei "/modules/shop/public/email_partner.php" (Name der Funktion, Beschreibung, Parameter):

shop_partner_freemail()	Email an einen Partner mit der Freischaltung und seinen Zugangsdaten.	$number
shop_partner_mail()	Email an einen Partner mit der Benachrichtigung über eine neue Bestellung.	$partner

6.9.436. Datei "/modules/shop/public/email_register.php"

Funktionen in der Datei "/modules/shop/public/email_register.php" (Name der Funktion, Beschreibung, Parameter):

shop_register_mail()	Email für den Kunden nach der Registrierung mit Informationen über den Zugang zum Kunden-Konto.	$array
shop_register_mail_admin ()	Email für den Shop-Betreiber nach der Registrierung mit Informationen zum neuen Kunden.	$number

6.9.437. Datei "/modules/shop/public/email_reviews.php"

Funktionen in der Datei "/modules/shop/public/email_reviews.php" (Name der Funktion, Beschreibung, Parameter):

shop_reviews_mail()	Email für den Shop-Betreiber mit der Benachrichtigung über eine neue Kundenmeinung.	$array

6.9.438. Datei "/modules/shop/public/email_rowisoft.php"

Funktionen in der Datei "/modules/shop/public/email_rowisoft.php" (Name der Funktion, Beschreibung, Parameter):

shop_rowisoft_mail()	Email für den Shop-Betreiber für den Import der Bestelldaten in die Warenwirtschaft Rowisoft blue.	$order

6.9.439. Datei "/modules/shop/public/info.php"

Funktionen in der Datei "/modules/shop/public/info.php" (Name der Funktion, Beschreibung, Parameter):

shop_show_info()	In dieser Datei wird über ein Array festgelegt, in welchen Dateien die über "index.php?file=shop&mode=info&type=[TYP]" abgerufenen Inhalte gespeichert sind (ausgehend vom Ordner "modules/shop/templates/").	$typ

6.9.440. Datei "/modules/shop/public/live_search.php"

Funktionen in der Datei "/modules/shop/public/live_search.php" (Name der Funktion, Beschreibung, Parameter):

shop_live_search()	Sucht passende Artikel noch während der Eingabe eines Suchbegriffes heraus.	$form

6.9.441. Datei "/modules/shop/public/login.php"

Funktionen in der Datei "/modules/shop/public/login.php" (Name der Funktion, Beschreibung, Parameter):

shop_login()	Zeigt das Formular für den Login bzw. bei eingeloggten Kunden das Kunden-Konto an.	
shop_set_login()	Login eines Kunden in das Kunden-Konto.	$form

6.9.442. Datei "/modules/shop/public/logout.php"

Funktionen in der Datei "/modules/shop/public/logout.php" (Name der Funktion, Beschreibung, Parameter):

shop_logout()	Logout eines Kunden aus dem Kunden-Konto.	$form

6.9.443. Datei "/modules/shop/public/main.php"

Funktionen in der Datei "/modules/shop/public/main.php" (Name der Funktion, Beschreibung, Parameter):

shop_main()	Hauptseite des öffentlichen Bereiches des Shops mit der Auflistung von Kategorien und Artikeln.	$section, $form

6.9.444. Datei "/modules/shop/public/partner.php"

Funktionen in der Datei "/modules/shop/public/partner.php" (Name der Funktion, Beschreibung, Parameter):

shop_partner()	Erste Seite der Partneranmeldung mit dem Fragebogen.	$form
shop_partner_2()	Zweite Seite der Partneranmeldung mit dem Formular für die Kontaktangaben.	$form
shop_partner_account()	Partner-Bereich mit Statistiken, Änderung der Partner-Daten usw.	$form
shop_partner_ads()	Zeit die Sonderseite mit den Werbemitteln für die Partner an.	$form
shop_partner_login()	Zeigt das Formular für den Login bzw. bei eingeloggten Partnern den Partnerbereich an.	$form
shop_partner_logout()	Logout eines Partner aus dem Partner-Konto.	$form
shop_partner_message ()	Zeigt ein Formular für den Versand einer Email an den Shop-Betreiber an.	$form
shop_partner_message_2 ()	Versendet die Email des Partners an den Shop-Betreiber.	$form
shop_partner_options()	Zeigt ein Formular für die Aktualisierung der Persönlichen Daten an.	$form
shop_partner_options_2 ()	Führt die Aktualisierung der Persönlichen Daten des Partners im Shop durch.	$form
shop_partner_order()	Zeigt die bisherigen Provisionszahlungen eines Partners an.	$form
shop_partner_password ()	Zeigt ein Formular für die Aktualisierung des Passwortes durch den Partner an.	$form
shop_partner_password_2 ()	Aktualisiert die Angaben zum Passwort eines Partners im Partner-Bereich.	$form
shop_partner_payment ()	Zeigt ein Formular für die Aktualisierung der Bankdaten an.	$form
shop_partner_payment_2 ()	Führt die Aktualisierung der Bankdaten des Partners im Shop durch.	$form
shop_partner_recovery ()	Zeigt ein Formular für "Passwort vergessen?" an und schickt dem Partner ein neues Passwort zu.	$form
shop_partner_send()	Fügt einen neuen Partner in die Datenbank ein und informiert diesen per Email über das Partnerprogramm.	$form
shop_partner_statistic()	Zeigt die Statistiken über Klicks / Verkäufe eines Partners an.	$form
shop_partner_success()	Gibt eine Meldung an den Partner aus, wenn die Anmeldung erfolgreich war.	$form
shop_set_partner_login ()	Login eines Partners in das Partner-Konto.	$form

6.9.445. Datei "/modules/shop/public/priceinfo.php"

Funktionen in der Datei "/modules/shop/public/priceinfo.php" (Name der Funktion, Beschreibung, Parameter):

shop_priceinfo()	Versendet die Anforderung eines Wunschpreises an den Shop-Betreiber.	$form

6.9.446. Datei "/modules/shop/public/random.php"

Funktionen in der Datei "/modules/shop/public/random.php" (Name der Funktion, Beschreibung, Parameter):

shop_random()	Zeigt einen zufälligen Artikel des Shops an.	

6.9.447. Datei "/modules/shop/public/recommend.php"

Funktionen in der Datei "/modules/shop/public/recommend.php" (Name der Funktion, Beschreibung, Parameter):

shop_recommend()	Zeigt das Formular für die Weiterempfehlung von Artikeln an.	$number
shop_recommend_now ()	Versendet auf der Basis der vorherigen Angaben eine Weiterempfehlung zu einem Artikel.	$form

6.9.448. Datei "/modules/shop/public/recovery.php"

Funktionen in der Datei "/modules/shop/public/recovery.php" (Name der Funktion, Beschreibung, Parameter):

shop_recovery()	Zeigt ein Formular für "Passwort vergessen?" an und schickt dem Kunden ein neues Passwort zu.	$form

6.9.449. Datei "/modules/shop/public/register.php"

Funktionen in der Datei "/modules/shop/public/register.php" (Name der Funktion, Beschreibung, Parameter):

shop_register()	Zeigt das Formular zur Eingabe der Kundendaten für die Registrierung an.	$form
shop_register_error()	Ausgabe einer Meldung an den Besucher über eine nicht erfolgreiche Registrierung.	
shop_register_now()	Führt die Registrierung eines neuen Kunden im Online-Shop durch.	$form
shop_register_success()	Ausgabe einer Meldung an den neuen Kunden über eine erfolgreiche Registrierung.	

6.9.450. Datei "/modules/shop/public/return.php"

Funktionen in der Datei "/modules/shop/public/return.php" (Name der Funktion, Beschreibung, Parameter):

shop_return()	Versendet die RMA-/Rückgabe-Anforderungen an den Shop-Betreiber.	$form

6.9.451. Datei "/modules/shop/public/reviews.php"

Funktionen in der Datei "/modules/shop/public/reviews.php" (Name der Funktion, Beschreibung, Parameter):

shop_add_review()	Fügt eine neue Kundenmeinung zu einem Artikel in die Datenbank ein.	$form

6.9.452. Datei "/modules/shop/public/search.php"

Funktionen in der Datei "/modules/shop/public/search.php" (Name der Funktion, Beschreibung, Parameter):

shop_search()	Suchfunktion im öffentlichen Bereich des Shops.	$q, $form
shop_search_subsections ()	Ermittelt die ID's der Unterkategorien einer Kategorie.	$number

6.9.453. Datei "/modules/statistic/class.admin.php"

Funktionen in der Datei "/modules/statistic/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_statistic ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_main()	Erste Seite des Statistik-Modules (Referer, Grafik, Allgemeine Statistik, Zeitverlauf usw.).	
do_page_2()	Zweite Seite des Statistik-Modules (Hits nach Browser, Hits nach Betriebssystem / Gerät, Beliebteste Seiten, Häufigste Herkunftsländer der Besucher).	
do_page_3()	Dritte Seite des Statistik-Modules (Häufigste Suchbegriffe auf der Webseite, Neueste Anfragen über die Suchmaschinen, Häufigste Anfragen über die Suchmaschinen).	
do_page_4()	Vierte Seite des Statistik-Modules (Häufigste Verweise auf die Webseite).	
do_page_5()	Fünfte Seite des Statistik-Modules (Häufigste User-Agents, Häufigste Hostnames / Provider).	
do_page_6()	Sechste Seite des Statistik-Modules (Feedburner, Aufrufe durch Suchmaschinen und Spider, Externe Dienste).	
do_process()	Verarbeitet die neuen Einträge in der Datenbanktabelle *_stats_power und legt Ergebnisse in *_stats_misc ab.	
do_restart()	Setzt die Statistik eines Mandanten komplett zurück.	
do_set_options()	Speicherung und Aktualisierung der Einstellungen.	$form
do_show()	Zeigt eine einzelne Statistik an, Aufruf direkt über "/admin.php?file=statistic&mode=show&form[show] =...&clean=yes".	$form
install()	SQL- und Dateianweisungen zur Installation des Modules.	
statistic_fav_countries()	Anzeige der Statistik "Häufigste Herkunftsländer der Besucher".	$form = array()
statistic_fav_pages()	Anzeige der Statistik "Beliebteste Seiten".	$form = array()

statistic_feedburner()	Anzeige der Statistik "Feedburner-Aktiväten (Abonnenten / Hits)".	$form = array()
statistic_hits_browser()	Anzeige der Statistik "Hits nach Browser".	$form = array()
statistic_hits_days()	Anzeige der Statistik "Seitenaufrufe und Besucher der letzten "n" Tage".	$form = array()
statistic_hits_hours()	Anzeige der Statistik "Hits pro Stunde".	$form = array()
statistic_hits_months()	Anzeige der Statistik "Hits pro Monat".	$form = array()
statistic_hits_os()	Anzeige der Statistik "Hits nach Betriebssystem / Gerät".	$form = array()
statistic_hits_user()	Anzeige der Statistik "Seitenaufrufe pro Besucher der letzten "n" Tage".	$form = array()
statistic_hits_weekdays ()	Anzeige der Statistik "Hits pro Wochentag".	$form = array()
statistic_hostnames()	Anzeige der Statistik "Häufigste Hostnames (Provider / Internetanbieter)".	$form = array()
statistic_last_searchwords ()	Anzeige der Statistik "Neueste Anfragen über die Suchmaschinen".	$form = array()
statistic_referer_overview ()	Anzeige der Statistiken "Grafik der häufigsten Verweise" + "Top 10 Verweise".	$form = array()
statistic_spider()	Anzeige der Statistik "Aufrufe durch Suchmaschinen und Spider".	$form = array()
statistic_top_intern()	Anzeige der Statistik "Häufigste Suchbegriffe auf der Webseite".	$form = array()
statistic_top_referer()	Anzeige der Statistik "Häufigste Verweise auf die Webseite".	$form = array()
statistic_top_searchwords ()	Anzeige der Statistik "Häufigste Anfragen über die Suchmaschinen".	$form = array()
statistic_user_agents()	Anzeige der Statistik "Häufigste User-Agents (genutzte Browser / Betriebssysteme)".	$form = array()
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.454. Datei "/modules/statistic/class.base.php"

Funktionen in der Datei "/modules/statistic/class.base.php" (Name der Funktion, Beschreibung, Parameter):

google_charts()	Gibt einzelne Daten als grafische Diagramme über Google Charts aus (http://code.google.com/apis/charttools/).	$data
visualize()	Gibt einzelne Daten als grafische Diagramme aus.	$name, $data = array(), $fields = array(), $title = "", $options = array()

6.9.455. Datei "/modules/statistic/class.index.php"

Funktionen in der Datei "/modules/statistic/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_statistic ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.456. Datei "/modules/statistic/class.webservices.php"

Funktionen in der Datei "/modules/statistic/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.457. Datei "/modules/statistic/container.php"

Funktionen in der Datei "/modules/statistic/container.php" (Name der Funktion, Beschreibung, Parameter):

statistic_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Für das Statstik-Modul ist dies die Ausgabe von Daten zu Besuchern, Seitenaufrufen usw. Aufruf in den Vorlagen: {cp:container module="statistic" [visitors="total"]} etc.	$form

6.9.458. Datei "/modules/tasks/class.admin.php"

Funktionen in der Datei "/modules/tasks/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_tasks ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_category()	Formular für das Hinzufügen einer neuen Kategorie.	

do_add_category_2()	Eine neue Kategorie wird in die Datenbank eingefügt.	$form
do_add_comment()	Ein Kommentar zu einer Aufgabe wird in die Datenbank eingefügt.	$form
do_add_task()	Formular für das Hinzufügen einer neuen Aufgabe.	
do_add_task_2()	Eine neue Aufgabe wird in die Datenbank eingefügt.	$form
do_delete_category()	Eine vorhandene Kategorie wird samt der zugeordneten Aufgaben und Unterkategorien gelöscht.	$number
do_delete_comment()	Ein vorhandener Kommentar zu einer Aufgabe wird gelöscht.	$number, $task
do_delete_task()	Eine vorhandene Aufgabe wird samt Kommentaren gelöscht.	$number
do_edit_category()	Formular für die Bearbeitung einer vorhandenen Kategorie.	$number
do_edit_category_2()	Eine Kategorie für die Aufgaben wird aktualisiert.	$form
do_edit_comment()	Formular für die Bearbeitung eines vorhandenen Kommentares.	$number, $task
do_edit_comment_2()	Ein vorhandener Kommentar zu einer Aufgabe wird aktualisiert.	$form
do_edit_task()	Formular für die Bearbeitung einer vorhandenen Aufgabe.	$number
do_edit_task_2()	Eine vorhandene Aufgabe wird aktualisiert.	$form
do_main()	Hauptansicht für das Modul Aufgaben mit einzelnen Formularen.	$order, $form, $month, $year
do_navigation()	Gibt einen Info-Text mit der aktuellen Anzahl an Aufgaben aus.	
do_search_now()	Leitet vom Suchformular auf die eigentliche Suche weiter.	$form
do_show_task()	Eine vorhandene Aufgabe wird samt Kommentaren angezeigt.	$number, $cdate
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.459. Datei "/modules/tasks/class.base.php"

Funktionen in der Datei "/modules/tasks/class.base.php" (Name der Funktion, Beschreibung, Parameter):

do_email_task()	Versand einer Email über neue Aufgaben an die jeweils berechtigten Benutzer.	$number, $to_email

6.9.460. Datei "/modules/tasks/class.index.php"

Funktionen in der Datei "/modules/tasks/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_tasks ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.461. Datei "/modules/tasks/class.webservices.php"

Funktionen in der Datei "/modules/tasks/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.462. Datei "/modules/tasks/container.php"

Funktionen in der Datei "/modules/tasks/container.php" (Name der Funktion, Beschreibung, Parameter):

tasks_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="tasks" [...]}	$forn

6.9.463. Datei "/modules/upload/class.admin.php"

Funktionen in der Datei "/modules/upload/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_upload ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_file()	Trifft Vorbereitungen für das Hinzufügen neuer Dateien.	$form
do_add_file_2()	Zeigt die Editoren für die Erstellung neuer Dateien an.	$form
do_add_file_3()	Fügt die neuen Dateien in das Dateisystem ein.	$articledesc, $form
do_blocked_contact()	Zeigt ein Formular zum Versenden von Kontaktanfragen bei blockierten Dateien an.	$object
do_blocked_contact_2()	Verschickt eine Email, wenn eine Datei vor Zugriffen geschützt ist.	$form
do_error()	Zeigt abhängig vom Typ verschiedene Fehlermeldungen an.	$type, $form = ""
do_error_file()	Gibt eine Fehlermeldung beim Upload einer Datei aus.	$object
do_file_search()	Ermöglicht die Suche nach Inhalten innerhalb der Dateien.	$form
do_file_search_now()	Leitet vom Suchformular auf die eigentliche Suche weiter.	$form
do_file_upload()	Lädt Dateien auf den Server und gibt eine Statusmeldung aus.	$path
do_getdir()	Zeigt eine Liste der Unterordner in einem Ordner an.	$sPath
do_getfile()	Zeigt eine Liste der Dateien in einem Ordner an.	$sPath
do_getsize()	Kalkuliert die Größe der in einem Ordner enthaltenen Dateien, inklusive der Dateien der Unterordner.	$directory

do_main()	Hauptansicht für das Modul Dateiverwaltung mit einzelnen Formularen.	$path
do_meta_edit()	Führt Änderungen wie Löschen und Bearbeiten an mehreren Dateien gleichzeitig durch.	$checkbox, $path, $setaction
do_mkdir()	Legt ein neues Verzeichnis im Dateisystem an.	$form
do_move_file()	Zeigt das Formular zum Verschieben von Dateien an.	$form
do_move_file_2()	Verschiebt eine Datei in ein neues Verzeichnis.	$form
do_options()	Zeigt die formatspezifischen Informationen zu einer Datei an.	$object
do_recursive()	Startet das rekursive Löschen, wenn diese Einstellung aktiviert ist.	$form
do_recursive_delete()	Löscht Dateien und Verzeichnisse rekursiv.	$root, $sub
do_rename_dir()	Zeigt das Formular zum Umbenennen von Ordnern an.	$form
do_rename_dir_2()	Führt die Umbenennung eines Ordners durch.	$form
do_rename_file()	Zeigt das Formular zum Umbenennen von Dateien an.	$form
do_rename_file_2()	Führt die Umbenennung einer bestehenden Datei durch.	$form
do_rmdir()	Führt die Umbenennung eines bestehenden Verzeichnisses durch.	$form
do_search()	Zentrale Suchfunktion für die einzelnen Unterordner des Dateisystems.	$form
do_search_now()	Leitet vom Suchformular auf die eigentliche Suche weiter.	$form
do_shortnavigation()	Zeigt die Kurznavigation auf der linken Seite an.	
do_text()	Zeigt den Text-Editor zum Bearbeiten einer Datei an.	$object
do_text_edit()	Speichert den Inhalt des Text-Editors in einer Datei ab.	$form
do_unlink()	Entfernt eine vorhandene Datei aus dem Dateisystem.	$form
do_upload_filesize()	Formatiert die Größe einer einzelnen Datei.	$filesize
do_wysiwyg()	Zeigt den WYSIWYG-Editor für die Bearbeitung an.	$object, $back = ""
do_wysiwyg_2()	Speichert den Inhalt des WYSIWYG-Editors in einer Datei ab.	$articledesc, $object, $back = ""
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.464. Datei "/modules/upload/class.index.php"

Funktionen in der Datei "/modules/upload/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_upload ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.
metatags()	Behandelt die Einblendung der Metatags.
search()	Stellt die Verbindung zur Volltextsuche her.

6.9.465. Datei "/modules/upload/class.webservices.php"

Funktionen in der Datei "/modules/upload/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.466. Datei "/modules/upload/container.php"

Funktionen in der Datei "/modules/upload/container.php" (Name der Funktion, Beschreibung, Parameter):

upload_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Aufruf in den Vorlagen: {cp:container module="upload" [...]}	$form

6.9.467. Datei "/modules/user/class.admin.php"

Funktionen in der Datei "/modules/user/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_user()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_group()	Fügt eine neue Benutzergruppe zur Datenbank hinzu.	$form, $form_rights
do_add_right()	Fügt ein neues Benutzerrecht zur Datenbank hinzu.	$form
do_add_user()	Zeigt das Formular zum Hinzufügen eines neuen Benutzer an.	
do_add_user_2()	Fügt einen neuen Benutzer zur Datenbank hinzu.	$form
do_choose_user()	Ermöglicht eine gleichzeitige Änderung mehrerer Benutzer (z.B. Änderung der Gruppenzuordnung).	$action, $checkbox
do_community()	Ermöglicht die Verwaltung der Gruppen und Einstellungen der Community / des Social Networks.	
do_delete_avatar()	Entfernt die Avatar-Grafik eines Benutzers aus der Datenbank und vom Server.	$number
do_delete_group()	Entfernt eine bestehende Benutzergruppe aus der Datenbank samt der dazugehörigen Verknüpfungen.	$number
do_delete_right()	Entfernt ein bestehendes Benutzerrecht aus der Datenbank.	$number
do_delete_user()	Entfernt einen bestehenden Benutzer aus der Datenbank.	$number
do_edit_group()	Formular für die Bearbeitung einer bestehenden Benutzergruppe samt Benutzerrechten.	$number
do_edit_group_2()	Aktualisiert die Daten für eine bestehende Benutzergruppe in der Datenbank.	$form, $form_rights
do_edit_right()	Formular für die Bearbeitung eines bestehenden Benutzerrechtes.	$number
do_edit_right_2()	Aktualisiert die Daten eines bestehenden Benutzerrechtes in der Datenbank.	$form
do_edit_right_now()	Leitet auf das Formular zum Bearbeiten eines vorhandenen Benutzers weiter.	$number

do_edit_status()	Ändert die Zuordnung zu einer Benutzergruppe für einen bestehenden Benutzer.	$form
do_edit_user()	Zeigt das Formular zum Bearbeiten eines vorhandenen Benutzers an.	$number
do_edit_user_2()	Geänderte Daten eines bestehenden Benutzers werden in der Datenbank aktualisiert.	$form
do_edit_user_sites()	Ändert die Mandantenzuordnung dieses Benutzers.	$form
do_error()	Zeigt eine Fehlermeldung innerhalb der Benutzerverwaltung an.	$type
do_getdir()	Listet die Unterverzeichnisse in einem Verzeichnis auf (in diesem Fall für die Darstellung der verfügbaren Sprachen).	$sPath
do_grouprights()	Zeigt die detaillierte Rechteverwaltung mit der grafischen Auflistung an.	$module, $number
do_grouprights_array()	Liefert das Array mit den verfügbaren Rechten für die erweiterte Rechteverwaltung zurück.	
do_groups()	Zeigt die Benutzergruppen samt Formular zum Hinzufügen einer neuen Benutzergruppe an.	
do_main()	Hauptansicht für das Modul Benutzerverwaltung mit einzelnen Formularen.	$form
do_options()	Zeigt das Formular für die Bearbeitung der Einstellungen des Modules an.	
do_rights()	Zeigt die im System verfügbaren Benutzerrechte an.	
do_search()	Leitet vom Suchformular auf die eigentliche Ausgabe des Suchergebnisses weiter.	$form
do_send_message()	Versendet eine Email-Nachricht an einen einzelnen Benutzer.	$number
do_send_message_2()	Versendet eine Email-Nachricht an einen einzelnen Benutzer.	$form
do_set_groupright()	Setzt die Benutzer-/Gruppenrechte für die detaillierte Rechteverwaltung.	$form
do_set_options()	Speichert die Einstellungen für die Benutzerverwaltung ab.	$form
do_user_form()	Formular für die Bearbeitung eines bestehenden Benutzers.	$type, $number = ""
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.468. Datei "/modules/user/class.base.php"

Funktionen in der Datei "/modules/user/class.base.php" (Name der Funktion, Beschreibung, Parameter):

generate_file_path()	Erzeugt zentral die Pfade zum Benutzer-Profil, zum Login usw.	$type, $array = array()

6.9.469. Datei "/modules/user/class.index.php"

Funktionen in der Datei "/modules/user/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_user()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_account()	Begrüßung für einen eingeloggten Besucher beim Aufruf des Benutzerbereiches.	
do_delete_avatar()	Entfernt die Avatar-Grafik eines Benutzers aus der Datenbank und vom Server.	
do_delete_member()	Zeigt die Warnmeldung an, bevor ein Benutzeraccount endgültig gelöscht wird.	
do_delete_member_2()	Führt die Löschung eines Benutzers auf eigenen Wunsch durch.	
do_edit_member()	Zeigt das Formular für die Bearbeitung der eigenen Daten durch einen Benutzer an.	$number
do_edit_member_2()	Aktualisiert die eigenen Daten eines Benutzers in der Datenbank.	$form
do_error()	Ausgabe einer allgemeinen Fehlermeldung an einen Benutzer.	
do_form_memberlist()	Leitet von der Suche auf die eigentliche Auflistung der Mitglieder weiter.	$form
do_get_back()	Aktualisierung / Neugenerierung eines Passwortes für einen Benutzer.	$form
do_getdir()	Generiert einer Auswahlliste aus den verfügbaren Ordnern.	$sPath
do_graphic()	Erstellt den grafischen Sicherheitscode für den Login in den Administrationsbereich.	
do_main()	Hauptansicht des Modules mit der Auswahl der gewünschten Inhalte.	$module, $admin
do_memberlist()	Erstellt die Auflistung der im System angemeldeten Benutzer.	$form
do_profil()	Erstellt das öffentliche Profil für einen Benutzer.	$number
do_random_position()	Legt zufällige Positionsangaben in einem Array fest.	$position_array, $type
do_recovery()	Formular für die Anforderung eines vergessenen Passwortes durch einen Benutzer.	
do_register()	Formular für die Registrierung eines neuen Benutzers.	$module
do_register_full()	Registrierungsdaten werden überprüft und in die Datenbank eingetragen.	$form
frontend_editing()	Behandelt die Einblendung der Frontend-Editing-Leiste für berechtigte Benutzer.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.470. Datei "/modules/user/class.webservices.php"

Funktionen in der Datei "/modules/user/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.471. Datei "/modules/user/container.php"

Funktionen in der Datei "/modules/user/container.php" (Name der Funktion, Beschreibung, Parameter):

user_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Für die Benutzerverwaltung sind dies ein Login-Formular bzw. eine Optionsleiste für eingeloggte Benutzer, die Anzeige von Benutzerlisten und Details zu einem einzelnen	$forn

	Benutzer. Aufruf in den Vorlagen: {cp:container module="user" [...]}	

6.9.472. Datei "/modules/webservices/class.admin.php"

Funktionen in der Datei "/modules/webservices/class.admin.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_webservices ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_add_import()	Fügt einen Newsfeed zur Datenbank hinzu.	$form
do_delete_import()	Entfernt sowohl den Datenbankeintrag, als auch die Quelldatei eines Newsfeeds.	$number
do_download()	Ermöglicht den Download der Klip-Datei (für Klipfolio).	$type
do_edit_import()	Ermöglicht die Bearbeitung eines vorhandenen Newsfeeds.	$number
do_edit_import_2()	Führt die Aktualisierung eines Newsfeeds in der Datenbank durch.	$form
do_import()	Zeigt die Übersicht der importierten Newsfeeds an.	$form
do_import_form()	Zeigt das Formular für den Import von Newsfeeds etc. an.	$number = ""
do_main()	Hauptansicht für das Modul Webservices mit einzelnen Formularen.	
do_set_options()	Speicherung und Aktualisierung der Einstellungen.	$form
do_start_import()	Führt den Import eines XML-basierten Newsfeeds durch.	
do_status()	Ändert den aktuellen Status des Newsfeeds.	$number, $status
do_success()	Zeigt eine Ergebnismeldung nach dem Ende einer Aktion an.	$number
install()	SQL- und Dateianweisungen zur Installation des Modules.	
uninstall()	SQL- und Dateianweisungen zur Deinstallation des Modules.	

6.9.473. Datei "/modules/webservices/class.index.php"

Funktionen in der Datei "/modules/webservices/class.index.php" (Name der Funktion, Beschreibung, Parameter):

contentpapst_gui_webservices ()	Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.	
do_ajax_container()	Nimmt einen Containeraufruf per AJAX entgegen und gibt den entsprechenden Inhalt aus.	$hash, $form
do_main()	Umleitung auf die Startseite des Systems.	
do_type()	Abhängig vom angegebenen Typ wird die Ausgabe der Inhalte erzeugt.	
metatags()	Behandelt die Einblendung der Metatags.	
search()	Stellt die Verbindung zur Volltextsuche her.	

6.9.474. Datei "/modules/webservices/class.webservices.php"

Funktionen in der Datei "/modules/webservices/class.webservices.php" (Name der Funktion, Beschreibung, Parameter):

process()	Methode, dient der Verarbeitung der ankommenden Informationen.	$command, $form

6.9.475. Datei "/modules/webservices/container.php"

Funktionen in der Datei "/modules/webservices/container.php" (Name der Funktion, Beschreibung, Parameter):

webservices_container()	Über diese Datei wird die Bereitstellung der einzelnen Container geregelt. Dabei können sowohl Atom-, RDF-, RSS- als auch einfache XML-Dateien angezeigt werden. Aufruf in den Vorlagen: {cp:container module="webservices"] [...]}	$form

6.9.476. Datei "/modules/webservices/newsfeed_food.php"

Funktionen in der Datei "/modules/webservices/newsfeed_food.php" (Name der Funktion, Beschreibung, Parameter):

webservices_newsfeed_food ()	Export der News des Systems im Food-Format (für KlipFolio).	$return = ""

6.9.477. Datei "/modules/webservices/newsfeed_klip.php"

Funktionen in der Datei "/modules/webservices/newsfeed_klip.php" (Name der Funktion, Beschreibung, Parameter):

webservices_newsfeed_klip ()	Export der zentralen Einstellungen und Klip-Informationen (für KlipFolio).	$return = ""

6.9.478. Datei "/modules/webservices/newsfeed_xml.php"

Funktionen in der Datei "/modules/webservices/newsfeed_xml.php" (Name der Funktion, Beschreibung, Parameter):

webservices_newsfeed_xml ()	Export der News des Systems im XML-Format (als RDF).	$return = ""

6.9.479. Datei "/themes/backend/main.php"

Funktionen in der Datei "/themes/backend/main.php" (Name der Funktion, Beschreibung, Parameter):

cp_admin_header()	Erstellt den Kopfbereich eines Modules mit der Unternavigation.	$basics, $array, $rights
cp_admin_languages()	Erstellt aus den im Ordner "/languages/" vorhandenen Unterordnern das Menü für den Wechsel zwischen den einzelnen Sprachen (in Abhängigkeit vom aktuellen Mandanten).	
cp_admin_mainmenu()	Erstellt die Hauptnavigation oben mit den Dropdown-Menüs. Aufbau: Name des benötigten Rechtes, URL, Titel, Alt-Tag, Buchstaben-Nr. für Accesskey, Breite in Pixel, zusätzliche Breite für Safari.	
cp_admin_overview()	Erstellt das Menü auf der linken Seite für das aktuell ausgewählte Modul.	$array, $level = "0", $last_entry = "0", $std_directory = ""
cp_admin_rights()	Ermittelt einzelne Berechtigungen des Benutzers innerhalb des Administrationsbereiches.	$array
cp_admin_shortinfo()	Erstellt die Texte für die Kurzinformation für ein bestimmtes Modul.	
cp_admin_sites()	Ermittelt die im System vorhandenen und für diesen Benutzer freigeschaltenen Mandanten (d.h. die verwalteten Webseiten).	
cp_admin_submenu()	Erstellt die Unternavigation für ein bestimmtes Modul.	$array, $rights = array()
sitefooter()	Gibt den Seitenfuß nach "{cp:content}" in der Hauptvorlage aus.	
siteheader()	Generierung des Seitenkopfes im Administrationsbereich mit der Anzeige der aktuellen Aufgaben, der verfügbaren Sprachen etc.	

6.10. Grafik

Die Grafik-Klasse der Entwickler-API ermöglicht die interaktive Erstellung von Kreis- und Balkengrafiken für Statistiken, die Anwendung verschiedener Effekte auf die Bilder usw. Es muss vorher die Datei "/functions/cp.image.php" per "include()" oder "include_once()" eingebunden werden. Zur Anwendung ist eine neue Instanz der Klasse zu initialisieren.

6.10.1. add_watermark()

Mit "add_watermark()" kann ein Bild mit einem halbtransparenten Wasserzeichen versehen werden. Dazu muss zuerst über die API ein vorhandenes Bild geöffnet oder aber ein neues Bild angelegt werden. Anschließend wird dann über diese Methode eine PNG-Grafik über das vorgeladene Bild gelegt. Diese PNG-Grafik kann dabei bereits mit Transparenz-Effekten versehen sein.

```
$image->add_watermark("pfad/zur/grafik/watermark.png"); // fügt die Grafik "watermark.png" als Wasserzeichen hinzu
```

6.10.2. brightness_contrast()

Diese Funktion passt Helligkeit und Kontrast eines Bildes den gewünschten Werten an. Optimal wird die Funktion erst kurz vor dem Speichern eines Bildes genutzt, um letzte Feinabstimmungen vorzunehmen. Der Funktion müssen lediglich die Angaben zu Helligkeit und Kontrast als numerische Werte zwischen -100 und +100 übergeben werden:

```
$image->brightness_contrast("10", "-15"); // die Helligkeit wird um 10% erhöht, der Kontrast um 15% vermindert
```

6.10.3. colorize()

"colorize()" ist eine Methode, mit welcher Grafiken eingefärbt werden können. Der Methode müssen als Angaben eine Farbangabe (RGB, als Array) und eine Prozentangabe (Stärke der Einfärbung) übergeben werden.

```
$image->colorize(array(238,243,245),"50"); // färbt eine Grafik zu 50 % gelb ein
```

6.10.4. colorpalette()

Über die Methode "colorpalette()" lassen sich die in einer JPEG-Grafik am häufigsten eingesetzten Farben und somit die Farbpalette ermitteln. Die Methode liefert dabei ein Array zurück. Jeder Schlüssel des Arrays enthält serialisiert ein Array der Farben. Der Wert zeigt jeweils, wie häufig die Farbe in der Grafik vorhanden ist.

```
$image->colorpalette("test.jpg", "50"); // gibt die 50 häufigsten Farben der Grafik zurück
```

6.10.5. create()

Wenn kein bereits vorhandenes Bild geladen werden soll, dann kann mit der Methode "create" ein neues Bild angelegt werden. Dieser Methode müssen dazu Breite und Höhe des Bildes in Pixel und der gewünschte Bildtyp übergeben werden. Als Bildtyp können "jpg" und "png" gewählt werden. Sehr vereinzelt wird auch "gif" unterstützt (dies ist abhängig von der PHP-Konfiguration).

```
$image->create("300", "500", "jpg"); // erstellt ein Bild im JPEG-Format mit der Abmessung 300x500 Pixel
```

6.10.6. gd_version()

Einige spezifische Methoden/Funktionen zur Arbeit mit Grafiken stehen nur bei einer GD-Version > 2.0 zur Verfügung (z.B. "imagecreatetruecolor()"). Diese Methode überprüft, welche GD-Version vorhanden ist und gibt das Ergebnis als Array mit

den Werten "short" (Kurze Nummer, z.B. "2") sowie "complete" (komplette Version-Nummer, z.B. "2.0.15") und "truecolor" (Unterstützung für Truecolor-Bilder [0/1]) zurück.

```
$image->gd_version(); // liefert Informationen zur GD-Version zurück
```

6.10.7. get_colorcode()

Die Methode "get_colorcode()" ermöglicht die Umwandlung von Farbangaben in ein anderes Format. Übergeben werden dazu das Ausgangsformat, sowie die Angabe, ob es sich um ein Hex- oder Dezimalformat handelt. Die Rückgabe ist der entsprechend umgewandelte Wert. So lassen sich mit der Methode einfach RGB-Farbangaben in HTML-Farbangaben wandeln.

```
$image->get_colorcode("#0F07FF",1); // liefert eine RGB-Angabe als Array zurück
$image->get_colorcode(array("120","86","42"),0); // liefert eine HTML-Angabe als String
```

6.10.8. grayscale()

Ähnlich wie bei der Methode "monochrome()" wird auch mit "grayscale()" die Anzahl der Farben eines Bildes reduziert. Allerdings sind hier zusätzlich zu Schwarz und Weiss noch verschiedene Grautöne enthalten. Gespeichert werden sollten derartige Bilder generell im PNG- oder GIF-Format, aufgrund der geringeren Dateigrößen.

```
$image->grayscale(); // führt die Konvertierung durch
```

6.10.9. invert()

Diese Methode invertiert die Farben eines kompletten Bildes. Die Methode wird dabei ohne zusätzliche Parameter aufgerufen:

```
$image->invert();
```

6.10.10. monochrome()

Mit der Methode "monochrome()" wird eine Schwarz-Weiss-Version des aktuellen Bildes erzeugt. Diese verfügt über keinerlei Zwischenfarben. Gespeichert werden sollten derartige Bilder generell im PNG- oder GIF-Format, aufgrund der geringeren Dateigrößen.

```
$image->monochrome(); // führt die Konvertierung durch
```

6.10.11. open()

Wie sich aus dem Namen schon ableiten lässt, kann über die Methode "open()" ein vorhandenes Bild geöffnet werden. Es werden dabei die Dateiformate JPEG, PNG und (sehr vereinzelt) GIF unterstützt. Letzteres ist von der PHP-Konfiguration abhängig. Dieser Methode wird nur der Dateiname sowie der Pfad übergeben. Weitere Parameter sind nicht notwendig.

```
$image->open("pfad/zur/datei/grafik.jpg"); // öffnet die JPEG-Datei "grafik.jpg"
```

6.10.12. piechart()

Mit "piechart()" lassen sich Statistiken in Form eines kreisförmigen Diagrammes darstellen inkl. Beschriftungen. Die Methode erhält Parameter für den Radius, die Beschriftung und den Wert der einzelnen Elemente sowie die Farben. Letztere werden per Array im RGB-Format übergeben.

```
$image->piechart(200,
array('Montag','Dienstag','Mittwoch','Donnerstag','Freitag'),
array(760,430,1508,980,1220),
array(
    array('200','40','50'),
    array('50','60','70'),
    array('100','90','80'),
    array('130','140','150'),
    array('15','9','255'),
)
);
```

Zeigt eine Grafik mit Angaben zu einzelnen Wochentagen, beispielsweise die Einnahmen eines Einzelhändlers. Im letzten Array werden die Farben zugewiesen (R,G,B).

6.10.13. resize()

Die Änderung der Bildgröße ist über die Methode "resize()" möglich. Der Methode müssen dabei die gewünschte Breite und Höhe des Bildes in Pixel übergeben werden. Ein optionaler dritter Parameter ermöglicht die Größenänderung im "truecolor"-Modus. Standardmäßig ist dieser aktiviert, mit dem Wert "0" kann dieser deaktiviert werden. Der "truecolor"-Modus bietet eine bessere Bildqualität bei einer ungefähr gleichgroßen Dateigröße.

```
$image->resize("200", "400", "1"); // der aktuelle Bild wird auf 200x400 Pixel verkleinert (im "truecolor"-Modus)
```

6.10.14. rotate()

Für die Bildbearbeitung per Browser ist eine Methode zum Drehen von Bildern sehr nützlich. "rotate()" berechnet alle Pixel eines Bildes für eine bestimmte Gradanzahl neu und liefert das Bild zurück. Der Methode wird hierzu lediglich die Stärke der Drehung in Grad übergeben.

```
$image->rotate("93"); // aktuelles Bild um 20° drehen
```

6.10.15. save()

Damit nach den bisher auf das Bild angewandten Methoden dieses Bild auch letztendlich wieder als Datei gespeichert werden kann, muss die Methode "save()" genutzt werden. Wenn der bisherige Dateiname beibehalten werden soll, dann benötigt diese Methode keine Parameter. Bei einem neuen Dateinamen hingegen wird der Methode der gewünschte Dateiname inkl. des Pfades übergeben:

```
$image->save("neuer/ordner/dateiname_neu.jpg"); // speichert die Datei als "neuer/ordner/dateiname_neu.jpg"
```

6.10.16. show()

Mittels "show()" wird ein erstelltes Bild an den Browser ausgegeben. Hierzu muss dieses als Bild aufgerufen werden (über den IMG-Tag). Es wird automatisch der Typ des Bildes benutzt, welcher bei der Erstellung angegeben bzw. beim öffnen des Bildes erkannt wurde (unterstützt werden momentan nur JPG und PNG).

```
$image->show(); // gibt das Bild aus
```

6.11. Mail

Mit den Methoden dieser Klasse können HTML- oder Textemails zusammengestellt und versandt werden. Auch das Anhängen beliebiger Dateien oder der Versand an mehrere Personen in Form eines Newsletters ist möglich. Um diese Methoden in Module einzubinden, muss die Datei "/functions/cp.mail.php" per "include()" oder "include_once()" eingebunden werden.

6.11.1. Beispiel 1

Mit diesem Beispiel soll die Leistungsfähigkeit der Klasse demonstriert werden. Hier wird eine neue Email im HTML-Format zusammengestellt. Sie erhält einen Anhang und wird anschließend an verschiedene Empfänger (TO, CC, BCC) versendet.

```
<?
// ...
// Contentpapst-Framework
// ...

include("functions/cp.mail.php"); // Einbindung der Klasse
include("functions/cp.mime.php"); // Einbindung der MIME-Typen

$mail = new contentpapst_mail;

$mail->set_type("html"); // Typ wird auf HTML gesetzt
$mail->set_mime_types($cp_mime_types);

$mail->set_subject("Einführung der neuen Klasse"); // Titel der Email

$mail->set_to("harry@domain.de","Harry Solid"); // Empfängeremail und -name
$mail->set_from("contact@sandoba.de","sandoba.de medien agentur"); // Absenderemail und -name (TO)

$mail->add_cc(array("test_a@sandoba.de","test_b@sandoba.de")); // CC-Empfänger in Array-Form
$mail->add_bcc(array("test_c@sandoba.de","test_d@sandoba.de")); // BCC-Empfänger in Array-Form

$mail->set_ip(); // IP-Adresse des Versenders wird integriert
$mail->set_priority(3); // Priorität normal
$mail->set_reply_to("reply_to@sandoba.de"); // Antworten gehen an diese Email-Adresse

$content = "Dies ist ein kleiner "TE<em>ST</em>"!!! ;-)"; // HTML-Inhalt der Email
$mail->set_body($content);

$mail->receive_receipt(); // Lesebestätigung anfordern

$mail->add_attachment("ordner/bild.gif","","","inline"); // Bild anfügen (wird in der Email angezeigt)
$mail->add_attachment("ordner/tool.zip"); // Datei anhängen

$mail->set_x_mailer($systemname." ".$systemversion." (".$systemurl.")");
$mail->set_x_bounced($systemname." ".$systemversion);

$mail->send(); // Email wird versandt
?>
```

6.11.2. Funktionsübersicht dieser Datei

In der Klasse "contentpapst_mail" in der Datei "/functions/cp.mail.php" sind u.a. die folgenden Funktionen/Methoden für die Erstellung und Bearbeitung von Emails vorhanden:

add_attachment	Dateianhang zur Email hinzufügen.
add_bcc	Einzelne Email-Adresse oder ein Array als "Blind-Carbon-Copy" (BCC) hinzufügen.
add_cc	Einzelne Email-Adresse oder ein Array als "Carbon-Copy" (BCC) hinzufügen.
create_header	Header der Email generieren.
get_bcc	Liefert die bisher als "Blind-Carbon-Copy" hinzugefügten Email-Adressen zurück.
get_cc	Liefert die bisher als "Carbon-Copy" hinzugefügten Email-Adressen zurück.
get_attachment	Liefert den Inhalt einer Datei aus dem Anhang zurück.
receive_receipt	Empfangsbestätigung für die Email anfordern.
send	Email versenden.
send_smtp	Email per SMTP versenden.
show_header	Header der Email anzeigen (nachdem "show_header" aufgerufen wurde).
set_bcc	Einzelne Email-Adresse als "Blind-Carbon-Copy" (BCC) festlegen.
set_cc	Einzelne Email-Adresse als "Carbon-Copy" (CC) festlegen.
set_charset	Zeichensatz der Email festlegen.
set_from	Email-Adresse des Absenders festlegen.
set_options	Verschiedene Optionen der Email festlegen.
set_ip	IP-Adresse des Absenders für den Email-Header festlegen.
set_mime_types	MIME-Typen der Email-Anhänge festlegen.
set_subject	Titel der Email festlegen.
set_priority	Priorität der Email festlegen (1-5).
set_reply_to	Antwort-Adresse für die Email festlegen.
set_to	Empfänger-Adresse für die Email festlegen.
set_type	Typ der Email festlegen ("text" oder "html").
validate_email	Überprüft, ob eine Email-Adresse valide ist.

6.12. RTF

Mit den Methoden dieser Klasse können vorhandene RTF-Dateien ausgelesen und umgewandelt werden. Auch das Anlegen neuer RTF-Dateien ist möglich. Um diese Methoden in Module einzubinden, muss die Datei "/functions/cp.rtf.php" per "include()" oder "include_once()" eingebunden werden.

6.12.1. Beispiel 1 (RTF-Datei anlegen)

In diesem Beispiel wird Ihnen gezeigt, wie Sie eine RFT-Datei anlegen und mit Inhalten füllen können. Nach der Erzeugung einer neuen Instanz der Klasse werden die notwendigen Daten übergeben, automatisch aufbereitet und als RTF gepeichert.

```
<?
// ...
// Contentpapst-Framework
// ...

include("functions/cp.rtf.php");
$rtf = new contentpapst_rtf;

$rtf->set_title("Meine RTF-Datei");
$rtf->set_author("Max Mustermann");
$rtf->set_company("Mustermann Ltd.");

$rtf->set_create_time(); // Erstellungsdatum in Sekunden (im Timestamp-Format)
$rtf->set_rev_time(time()-20000); // Datum der letzten Bearbeitung

$rtf->set_content("Hier können Sie <b>beliebige HTML- oder Text-Inhalte<<b> einfügen, alle <font
size="25">unterstützten <em>Tags</em></font> werden umgewandelt, andere automatisch
<strike>gelöscht</strike> entfernt.<br/><br/>So sind Sie nicht nur auf einfache Textformierungen beschränkt...");

$content = $rtf->set_capitals(); // Alles in Großbuchstaben (ja/nein)

// RTF erzeugen und als "myrtf.rtf" abspeichern

$rtf->create();
$rtf->save("myrtf.rtf");
?>
```

6.12.2. Beispiel 2 (RTF-Datei konvertieren)

Hier wird eine (zuvor erzeugte) RTF-Datei eingelesen und verarbeitet. Wichtige RTF-Formatierungen werden dabei nach Möglichkeit in HTML konvertiert.

```
<?
// ...
// Contentpapst-Framework
// ...

include("functions/cp.rtf.php");
$convert = new contentpapst_rtf;

$convert->open("myrtf.rtf");
$convert->set_content();
$convert->convert();
?>
```

6.13. Sessions - Temporäre Benutzerinformationen

Über die Funktionen in der Datei "/functions/cp.sessions.php" können temporäre Benutzerinformationen angelegt und später wieder gelöscht werden. Diese Funktionen kommen insbesondere bei geschützten Bereichen zum Einsatz, um die Login-Daten des Benutzers zu speichern, so dass innerhalb eines Bereiches kein wiederholter Login notwendig ist.

6.13.1. cp_delete_session()

Wenn Session-Informationen nicht mehr benötigt werden, dann können diese über "cp_set_session()" erstellten Informationen mit "cp_delete_session()" wieder gelöscht werden. Dazu muss der Funktion der Name dieser Benutzerinformation übergeben werden:

```
cp_delete_session("log_password"); // Löscht die Benutzerinformation "log_password"
```

6.13.2. cp_set_session()

Über die Funktion "cp_set_session()" kann eine Benutzerinformation in einer Session abgelegt werden. Dazu muss der Funktion der Name der Benutzerinformation sowie der Wert übergeben werden:

```
cp_set_session("log_password", $password); // Wert der Variable "$password" wird als "log_password" gespeichert
```

Diese Information ist dann bis zum Schließen des Fensters über *<?php echo $_SESSION["log_password"]; ?>* abrufbar.

6.14. Statistik

Die Angaben im Statistik-Modul werden mit Hilfe der Funktionen der Datei "/functions/cp.statistic.php" generiert. Über diese Funktionen können u.a. der Browser, das Betriebssystem und die Spracheinstellungen eines Besuchers ermittelt werden.

6.14.1. cp_statistic_browser()

Die Funktion "cp_statistic_browser()" liefert Angaben zum genutzten Browser des Besuchers zurück. Mögliche Rückgabewerte sind u.a. Internet Explorer, Opera, Konquerer, Netscape, Firefox und Mozilla. Der Funktion muss zur Überprüfung der User-Agent des Besuchers übergeben. "cp_statistic_browser()" wird standardmäßig innerhalb der Funktion "cp_statistic_info()" mit ausgewertet.

```
$browser = cp_statistic_browser(strtolower($_SERVER["HTTP_USER_AGENT"])); // liefert den Namen des Browsers zurück
```

6.14.2. cp_statistic_info()

Wenn umfangreichere Angaben zu einem Benutzer benötigt werden, dann bietet sich die Benutzung der Funktion "cp_statistic_info" an. Auf diesem Wege müssen die Funktionen der Statistik nicht einzeln aufgerufen werden. Diese Funktion liefert in einem Array Angaben zum Browser, zum Betriebssystem, zur Javascript-Unterstützung, zum Referer und zur Standard-Sprache des Benutzers zurück:

```
$array = cp_statistic_info();
```

6.14.3. cp_statistic_isbot()

Mit dieser Funktion kann überprüft werden, ob es sich bei dem Besucher um einen Menschen handelt oder um ein Script wie z.B. den Crawler einer Suchmaschine. Dazu wird der User-Agent des Besuchers nach verschiedenen Begriffen überprüft. U.a. können mit Hilfe dieser Funktion einzelne Bereiche der Webseite vor dem Zugriff durch Suchmaschinen-Crawler geschützt werden. Mögliche Rückgabewerte der Funktion sind "0" (false) und "1" (true).

```
$is_bot = cp_statistic_isbot(strtolower($_SERVER["HTTP_USER_AGENT"])); // liefert das Ergebnis der Überprüfung zurück
```

6.14.4. cp_statistic_javascript()

Diese Funktion überprüft, ob der Browser des Benutzers auch prinzipiell Javascript-fähig ist. Es kann allerdings nicht

geprüft werden, ob auch tatsächlich Javascript aktiviert ist. Als Parameter muss der Funktion dabei das Ergebnis der Funktion "cp_statistic_browser()" übergeben werden. Mögliche Rückgabewerte sind "0" (false) und "1" (true).

```
$browser = cp_statistic_browser(strtolower($_SERVER["HTTP_USER_AGENT"])); // liefert den Namen des Browsers zurück
$js = cp_statistic_javascript(strtolower($browser));
```

6.14.5. cp_statistic_language()

Über diese Funktion kann die Standard-Sprache eines Besuchers ermittelt werden. Dazu werden der Funktion auf der IP-Adresse des Benutzers basierende Informationen übergeben. Allgemein wird diese Funktion daher nur innerhalb der Funktionen "cp_statistic_info" und "cp_statistic" aufgerufen. Sofern eine Standard-Sprache ermittelt werden konnte, wird diese in Form des Länderkürzels zurückgeliefert (also z.B. "fr" für "Französisch").

6.14.6. cp_statistic_os()

Die Funktion "cp_statistic_os()" liefert Angaben zum genutzten Betriebssystem des Besuchers zurück. Mögliche Rückgabewerte sind u.a. Windows, Linux, Macintosh, FreeBSD, UNIX und AIX. Der Funktion muss zur Überprüfung der User-Agent des Besuchers übergeben. "cp_statistic_os()" wird standardmäßig innerhalb der Funktion "cp_statistic_info()" mit ausgewertet.

```
$os = cp_statistic_os(strtolower($_SERVER["HTTP_USER_AGENT"])); // liefert den Namen des Betriebssystems zurück
```

6.15. URL

Diese Funktionen ermöglichen beispielsweise das Auslesen von Titel und Metatags einer Webseite oder die Auflösung der URL in einzelne Teile. Um die Funktionen in Module einzubinden, muss die Datei "/functions/cp.url.php" per "include()" oder "include_once()" eingebunden werden.

6.15.1. cp_url_element()

Diese Funktion liefert ein bestimmtes Element aus einer vorgegebenen URL zurück. Als Element können "scheme", "host", "port", "user", "pass", "path", "query", und "fragment" genutzt werden.

```
cp_url_element("http://user@www.url.de:80/ordner/datei.php?variable=15","query")
```

Verfügbare Elemente einer URL	
scheme	Schema / Protokoll ("http")
host	Host- / Domainname ("www.url.de")
port	Port ("80")
user	Benutzername ("rot")
pass	Passwort
path	Pfad ("/ordner/datei.php")
query	Parameter ("variable=15")
fragment	Fragmente

6.15.2. cp_url_frameset()

Eine weitere Funktion zur Zerlegung von HTML-Seiten ist "cp_url_frameset()". Mit dieser Funktion lassen sich Framesets in die einzelnen Framequellen aufteilen. Alle Seiten, zu denen das Frameset verlinkt, werden in einem Array abgelegt mit den möglichen Attributen "href", "name" und "scolling". Auch hier ist die direkte Übergabe von Daten möglich.

```
$var = cp_url_frameset("frameset.html"); // Frame-URLs in "frameset.html" ermitteln
```

6.15.3. cp_url_header()

Der Funktion "cp_url_header()" wird eine URL übermittelt. Daraufhin wird eine Verbindung zu dieser URL aufgebaut und der Header ausgelesen. Dieser Header wird anschließend zurückgeliefert. Werden die Daten als Array benötigt, so kann dies als zusätzlicher Parameter angegeben werden. Über weitere Funktionen können genauere Daten ausgelesen werden.

```
$var = cp_url_header("http://www.url.de"); // liefert den Header von "www.url.de"
$var = cp_url_header("http://www.url.de","array"); // liefert den Header von "www.url.de" als Array zurück
```

Zurückgegebene Werte (in Form eines Arrays / Strings)	
type	Verbindungs-Typ
version	HTTP-Version
status	HTTP-Statuscode
description	Meldung zum Statuscode
server	Serverangaben
date	Datum des Zugriffs
content-length	Länge der Datei in Bytes

content-type	MIME-Typ der Datei

6.15.4. cp_url_images()

Die Funktion "cp_url_images()" dient der Gewinnung von Grafiken von einer vorgegebenen URL. Dies kann z.B. für die Archivierung von ganzen Webseiten nützlich sein. Die Seite wird hierzu eingelesen und durchläuft eine Routine, welche die Grafiken herausfiltert und die einzelnen Attribute geordnet in einem Array ablegt. Alternativ kann auch direkt ein Text übergeben werden, indem ein zweiter Parameter mit dem Wert "1" angehängt wird.

Die Rückgabe ist ein Array, in welchem als Attribute eines Bildes "width", "height", "border", "src" und "alt" abgelegt sein können. Grafiken, welche mehrfach auftreten, werden ignoriert.

```
$var = cp_url_images("datei.htm"); // filtert alle Bilder aus "datei.htm"
```

6.15.5. cp_url_is_host()

Die Funktion "cp_url_is_host()" überprüft, ob eine Datei auf dem lokalen bzw. einem anderen Server liegt. Dies kann u.a. zur Sicherung bei der Arbeit mit Dateien genutzt werden. Der Funktion wird hierzu als Parameter die URL der Datei übergeben. Im Erfolgsfall (Datei ist lokal vorhanden) wird TRUE zurückgegeben, andernfalls FALSE.

```
$var = cp_url_is_host("http://www.yahoo.com/index.php"); // URL nicht auf lokalem Server vorhanden (Ergebnis: FALSE)
```

6.15.6. cp_url_links()

Die Funktion "cp_url_links()" dient der Gewinnung von Links von einer vorgegebenen URL. Dies kann z.B. für die Archivierung von ganzen Webseiten nützlich sein. Die Seite wird hierzu eingelesen und durchläuft eine Routine, welche die Links herausfiltert und die einzelnen Attribute geordnet in einem Array ablegt. Alternativ kann auch direkt ein Text übergeben werden, indem ein zweiter Parameter mit dem Wert "1" angehängt wird.

Die Rückgabe ist ein Array, in welchem als Attribute eines "href" und "target" abgelegt sein können. Links, welche mehrfach auftreten, werden ignoriert.

```
$var = cp_url_links("datei.htm"); // filtert alle Links aus "datei.htm"
```

6.15.7. cp_url_lookup()

Sie können mit der Funktion "cp_url_lookup()" überprüfen, ob eine bestimmte URL auch tatsächlich verfügbar ist. Intern wird auf die Funktionen "cp_url_extend()" und "cp_url_format()" zurückgegriffen, um die Konformität der URL zu gewährleisten.

```
$var = cp_url_lookup("http://www.url.de"); // testet, ob die URL "www.url.de" verfügbar ist
```

6.15.8. cp_url_metatags()

Bauen Sie einen eigenen Spider und / oder Link-Kataloge auf. Die Funktion "cp_url_metatags()" liest alle Meta-Tags aus einer vorgegebenen URL aus und liefert diese als Array zurück. Sind bestimmte Meta-Tags nicht vorhanden, wird kein Array-Schlüssel erzeugt.

```
$var = cp_url_metatags("datei.htm"); // liest die Meta-Tags von "datei.htm" aus
```

Ausnahmen sind "title" (wird ersetzt durch dem Titel der Webseite) und "description" (liest 300 Zeichen aus dem Textbereich der Webseite). Diese Meta-Tags werden ausgelesen:

Zurückgelieferte Meta-Tags (soweit enthalten)	
audience	Zielgruppe der Webseite (Erwachsene, Kinder usw.)
author	Autor/Webmaster der Seite
content_language	Sprache (nach ISO 639-1
content_type	MIME-Typ der Datei
copyright	Copyright-Inhaber der Webseite
date	Veröffentlichungsdatum
description	Beschreibung der Webseite (max. 1000 Zeichen)
distribution	Beschreibung des Verbreitungsgebietes
expires	Verfallsdatum
keywords	Stichwörter, Schlüsselwörter
language	Sprache (nach ISO 639-1)
page_topic	Stichwort für Seiteninhalt
pragma	Cache-Verhalten für Proxy-Server
publisher	Verantwortlicher für Veröffentlicung
refresh	Aktualisierung nach "x" Sekunden
resource_type	Typ der Webseite
revisit_after	Webseite alle "x" Tage besuchen

robots	Anweisung für Spider / Robots
title	Titel der Webseite

6.15.9. cp_url_port()

Um Ihren Webserver vor unerwünschten Zugriffen zu schützen, sollten Sie stets überwachen, welche Ports geöffnet oder geschlossen sind. Die Funktion "cp_url_port()" stellt das hierfür notwendige Hilfsmittel dar. Übermittelt werden neben der URL zum Webserver der Port sowie ein Zeitlimit (in Sekunden)

```
$var = cp_url_port("localhost",80,5); // überprüft den Port 80 auf "localhost"
```

6.15.10. cp_url_title()

Mit "cp_url_title()" ermitteln Sie den Titel/Namen einer Webseite, sofern dieser im Quelltext angegeben wurde. Der Funktion wird hierzu die gewünschte URL übergeben. Intern wird auf die Funktionen "cp_url_extend()" und "cp_url_format ()" zurückgegriffen, um die Konformität der URL zu gewährleisten.

```
$var = cp_url_title("http://de.yahoo.com/"); // gibt den Titel der Yahoo! Deutschland-Webseite aus
```

6.16. Verschiedenes

In dieser Funktionssammlung sind verschiedene Bestandteile der Entwickler-API enthalten, welche die Extrahierung von Dateiinformationen wie ID3v2, IPTC oder EXIF usw. ermöglich. Um diese Funktionen in Module einzubinden, muss die Datei "/functions/cp.misc.php" per "include()" oder "include_once()" eingebunden werden.

6.16.1. cp_clean_wysiwyg()

Dieser Funktion kommt in Contentpapst besondere Bedeutung zu. Mit "cp_clean_wysiwyg()" lässt sich ein per Variable übergebener Text aus dem WYSIWYG-Editor standardkonform aufbereiten. Es wird versucht, alle HTML-Tags in das XHTML-Format zu konvertieren.

```
$var = cp_clean_wysiwyg($string); // gibt den neuen Inhalt von "$string" zurück
```

6.16.2. cp_convert_filesize()

Verschiedene Dateigrößen können mit dieser Funktion in ein anderes Format konvertiert werden. Der Funktion wird neben der Dateigröße in Bytes das gewünschte Zielformat (KB, MB, GB, TB, PB, EB) sowie optional die Anzahl der Dezimalstellen übergeben.

```
$var = cp_convert_filesize("204922","KB","2"); // gibt gibt die Dateigröße in KB mit zwei Dezimalstellen zurück
```

6.16.3. cp_extract_exif()

Die Funktion "cp_extract_exif()" extrahiert die EXIF-Informationen in einer JPEG- oder TIFF-Datei und gibt diese als assoziatives Array zurück. EXIF enthält wichtige Angaben zu Grafiken wie die Bittiefe, die Verschlussgeschwindigkeit und das Erstellungsprogramm. Diese Angaben sind nicht immer zwingend für eine Grafik erforderlich und können daher variabel ausfallen.

EXIF-Informationen werden meist von Digitalkameras in die Grafiken eingefügt, damit eine weitergehende Anpassung der Grafiken anhand der Einstellungen bei der Aufnahme vorgenommen werden kann. Damit die Funktion arbeiten kann, wird ihr ein Dateiname samt Pfad ausgehend vom Contentpapst-Verzeichnis übergeben. Zurückgegeben werden folgende Angaben, sofern diese in einer Datei vorhanden sind:

Zurückgelieferte EXIF-Informationen (soweit enthalten)	
aperture_fnumber	Blendeneinstellung
aperture_value	Blenden-Wert in APEX
artist	Autor/Artist
brightness_value	Helligkeitswert der Grafik in APEX
color_space	Farbraum / Bittiefe des Bildes
copyright	Copyright-Informationen
date	Datum der Erstellung
date_time	Datum/Zeit der letzten Änderung
date_time_digitized	Datum/Uhrzeit der Digitalisierung
date_time_original	Datum/Uhrzeit des Originals
description	Beschreibung der Grafik
exif_image_length	Größe der EXIF-Grafik
exif_image_width	Größe der EXIF-Grafik
exif_version	Verwendete Version der EXIF-Spezifikation
exposure_bias_value	Exposure Bias-Wert in APEX
exposure_program	Genutztes Programm für die Datei
exposure_time	Verschlussgeschwindigkeit

file_source	Quelle der Datei (Digitalkamera etc.)
flash	Blitzlicht (0 = kein Blitzlicht, >= 1 = Blitzlicht)
flash_pix_version	Verwendete FlashPix-Version
fnumber	F-number (F-stop)
focal_length	Brennweite in Millimeter
focal_plane_resolution_unit	Abmessungen der Brennebene (1 = keine Einheit, 2 = Inch, 3 = Zentimeter)
focal_plane_x_resolution	Horizontale Auflösung der Brennebene
focal_plane_y_resolution	Vertikale Auflösung der Brenneben
height	Höhe der Grafik
is_color	Farbbild (0 = Nein, 1 = Ja)
iso_speed_ratings	Äquivalent zum CCD Index
make	Hersteller der Digitalkamera
metering_mode	Modus der Entfernungsmessung
mime_type	MIME-Type (z.B. "image/jpeg")
model	Modellbezeichnung der Digitalkamera
name	Name der Datei ohne Pfadangaben
orientation	Ausrichtung (1 = oben links, 2 = oben rechts usw.)
resolution_unit	Abmessungen (1 = keine Einheit, 2 = Inch, 3 = Zentimeter)
scene_type	Typ der Szene (direkt aufgenommen etc.)
sensing_method	Typ des genutzten Sensors der Digitalkamera
shutter_speed_value	Verschlussgeschwindigkeit in APEX
size	Größe in Bytes
size_formatted	Größe in MB umgerechnet
software	Software der Digitalkamera bzw. Bildbearbeitungsprogramm
status	Error-Code der Datei (1 = alles OK, 2 = Datei nicht vorhanden, 3 = falscher Dateityp, 4 = keine EXIF-Unterstützung vorhanden)
type	Dateityp
x_resolution	Breite der Grafik für den Druck (meist in Inch)
y_resolution	Höhe der Grafik für den Druck (meist in Inch)
width	Breite der Grafik
ycbcr_positioning	Positionierung der "YCbCr"-Elemente

Weitere Angaben können je nach Modell der Digitalkamera zurückgegeben werden, sind jedoch nicht standardisiert und daher nicht immer verlässlich.

```
$var = cp_extract_exif("images/regenwald.jpg"); // liefert alle Angaben zu dieser Datei zurück
```

6.16.4. cp_extract_id3()

Die Funktion "cp_extract_id3()" extrahiert die ID3- bzw. ID3v2-Informationen in einer MP3-Datei und gibt diese als assoziatives Array zurück. ID3/ID3v2 enthält wichtige Angaben zum Musikstück wie die Dauer, die Abtastrate und Bitrate, MPEG-Version und MPEG-Layer, aber auch zusätzliche Angaben wie Autor, Titel, einen Kommentar usw.

Damit die Funktion arbeiten kann, wird ihr ein Dateiname samt Pfad ausgehend vom Contentpapst-Verzeichnis übergeben. Zurückgegeben werden folgende Angaben, sofern diese in einer Datei vorhanden sind:

Zurückgelieferte ID3- bzw. ID3v2-Informationen (soweit enthalten)	
album	Album, aus dem die Datei stammt
artist	Autor/Artist des Musikstückes
bitrate	Bitrate der Datei in kBit/s
comment	Kommentar zum Musikstück
copyright	Copyright-Informationen (-1 = Nicht geschützt, 1 = Geschützt)
emphasis	Betonung des Musikstückes (keine einheitliche Nutzung)
framerate	Anzahl der Datenblöcke in der Datei
genre	Genre-Nummer
genre_name	Name des Genres
length	Gesamte Spieldauer in Sekunden
length_formatted	Spieldauer in lesbarer Form mit Minuten und Sekunden
length_minutes	Verbleibende Minuten der Spieldauer
length_seconds	Verbleibende Sekunden der Spieldauer
mode	Anzahl der genutzten Kanäle ("Stereo", "Joint-Stereo", "Dual-Channel" oder "Mono")
mode_extension	Nicht genutztes Bit
mpeg_layer	Der genutzte MPEG-Layer (Layer I/II/III)
mpeg_version	MPEG-Version der Datei (MPEG 1/2)
name	Name des Musikstückes ohne Pfadangaben
original_home	Modus (0 = Kopie, 1 = Original)
padding_bit	Dient der Berechnung der Frame-Anzahl
private_bit	Bit zur privaten Verfügung in einer Datei
protection_bit	Schutz-Bit am Anfang einer Datei (0 = Deaktiviert, 1 = Aktiviert)
sampling_frequenze	Abtastrate in Hz

size	Größe in Bytes
size_formatted	Größe in MB umgerechnet
status	Error-Code der Datei (1 = alles OK, 2 = Datei nicht vorhanden, 3 = falscher Dateityp)
syncword	Synchronisierungs-Bits am Anfang einer Datei
time	Erstellungsdatum der Datei
title	Songtitel

```
$var = cp_extract_id3("music/yvonne_cutterfeld_solo.mp3"); // liefert alle Angaben zu dieser Datei zurück
```

6.16.5. cp_extract_iptc()

IPTC ist ein Standard der "International Press Telecommunications Council" und erlaubt die zusätzliche Integration von beschreibenden Informationen in den Kopf-Bereich einer JPEG-Datei. Die Funktion "cp_extract_iptc()" extrahiert diese Informationen und gibt diese als assoziatives Array zurück. Die Angaben sind nicht immer zwingend für eine Grafik erforderlich und können daher variabel ausfallen.

Damit die Funktion arbeiten kann, wird ihr ein Dateiname samt Pfad ausgehend vom Contentpapst-Verzeichnis übergeben. Zurückgegeben werden folgende Angaben, sofern diese in einer Datei vorhanden sind:

Zurückgelieferte IPTC-Informationen (soweit enthalten)	
author	Autor
author_description	Autor der Beschreibung
caption	Bildbeschreibung
category	Kategorie
city	Stadt
copyright_notice	Copyright-Informationen
country	Land
creation_date	Erstellungsdatum
credit_byline_title	Position des Autors
headline	Überschrift
height	Höhe der Grafik
keywords	Stichwörter
name	Name der Grafik ohne Pfadangaben
otr	Aufgeber-Code
photo_source	Quelle
priority	Dringlichkeit
size	Größe in Bytes
size_formatted	Größe in MB umgerechnet
source	Bildrechte
spec_instr	Anweisungen
state	Bundesland
status	Error-Code der Datei (1 = alles OK, 2 = Datei nicht vorhanden)
supplemental_category	Ergänzende Kategorie
title	Bildtitel
width	Breite der Grafik

```
$var = cp_extract_iptc("images/regenwald.jpg"); // liefert alle Angaben zu dieser Datei zurück
```

6.16.6. cp_extract_pdf()

Die Funktion "cp_extract_pdf()" extrahiert einige Metadaten aus PDF-Dateien und gibt diese als assoziatives Array zurück. Mögliche Metadaten sind hierbei der Autor, das zur Erstellung verwendete Programm, Schlüsselwörter, das Erstellungsdatum, der Titel und ein Betreff. Damit die Funktion arbeiten kann, wird ihr ein Dateiname samt Pfad ausgehend vom Contentpapst-Verzeichnis übergeben.

```
$var = cp_extract_pdf("downloads/pdf/pressemitteilung-23-07-04.pdf"); // liefert alle Angaben zu dieser Datei zurück
```

6.16.7. cp_get_charset()

Ermitteln Sie den zu einer Sprache gehörigen Zeichensatz. Die Funktion erwartet den Namen einer Sprache im zweistelligen ISO 639-1-Format. Der einzige Rückgabewert ist das verwendete ISO-Format des Zeichensatzes.

```
$var = cp_get_charset("jp"); // gibt den Zeichensatz für Japanisch zurück ("iso-8859-1")
```

6.16.8. cp_js_alert()

Wenn Sie auf einfachem Wege eine Alarm-Meldung per Javascript ausgeben wollen, dann sollten Sie "cp_js_alert()" einsetzen. Per Parameter wird der auszugebende Text übergeben, ein zweiter Parameter gibt an, ob der resultierende Quellcode zurückgegeben ("1") oder direkt ausgeführt ("0") werden soll.

```
cp_js_alert("Hallo Welt!","0"); // hier wird direkt "Hallo Welt!" angezeigt
```

6.17. Vorlagen

Mit den Methoden dieser Klasse können Vorlagen erstellt, geladen und verarbeitet werden. U.a. werden mit diesen Funktionen die in den Vorlagen enthaltenen {cp:tags} durch die jeweils angeforderten Inhalte ersetzt. Um diese Methoden in Module einzubinden, muss die Datei "/functions/class.templates.php" per "include()" oder "include_once()" eingebunden werden.

6.17.1. Funktionsübersicht dieser Datei

In der Klasse "contentpapst_class_template" in der Datei "/functions/class.templates.php" sind u.a. die folgenden Funktionen/Methoden für die Erstellung und Verarbeitung von Vorlagen vorhanden:

assign	Funktion für die Zuweisung von Werten zu Variablen für "{cp:assign}".
change	Überprüfung von Variablenwerten für "{cp:if:*}" und "{cp:ifnot:*}".
clean	Löscht die bisher geladene Vorlage und setzt somit die Klasse zurück.
dump	Speichert eine verarbeite Vorlage als Datei (z.B. für den Cache).
format	Verarbeitet die Container ("{cp:container*}") und setzt die Inhalte ein.
format_client	Führt den Tag "{cp:client}" aus und überprüft u.a. Browser und Betriebssystem des Benutzers.
format_date	Führt den Tag "{cp:date*}" für Datumsangaben aus.
format_language	Führt die {cpl:tags} für die Sprachinhalte aus.
format_texteffect	Führt den Tag "{cp:texteffect*}" für Textformatierungen aus.
open	Öffnet eine neue Vorlage.
output	Startet die Verarbeitung der enthaltenen {cp:tags} und liefert den Inhalt der fertigen Vorlage zurück.
rewrite	Ersetzt einen übergebenen String durch das mod_rewrite-fähige Äquivalent (z.B. für Dateinamen).
process	Startet die Verarbeitung der {cp:tags} in den Vorlagen (u.a. über die oben gelisteten Funktionen.
replace	Ersetzt einen {cp:tag} durch den Inhalt einer Variablen.
rights	Führt den Tag "{cp:rights*}" für die Benutzerberechtigungen aus.

6.18. XML

Diese Funktionen ermöglichen die Interaktion mit XML-Inhalten bzw. -Dateien. Diese können eingelesen, zu Arrays verarbeitet und gespeichert werden. Um die Funktionen in Module einzubinden, muss die Datei "/functions/cp.xml.php" per "include()" oder "include_once()" eingebunden werden.

6.18.1. Beispiel 1

Dieses kurze Beispiel soll veranschaulichen, wie die XML-Funktionen von Contentpapst genutzt werden können. Zuerst wird eine vorhandene XML-Datei eingeladen.

Anschließend werden verschiedene Knoten hinzugefügt, entfernt und mit Inhalten gefüllt. Die bearbeitete Datei wird anschließend als XML-Datei neu gespeichert. Der Haupt-Knoten ist in diesem Fall "channel":

```
<?
// ...
// Contentpapst-Framework
// ...

include("functions/cp.xml.php");
$xml = new contentpapst_xml; $xml->open("text.xml");
$xml->add_node("channel/0/item/$a","options");
$xml->remove_node("channel/0/item/$a/link");
$xml->remove_content("channel/0/item/$a/title");
$xml->set_content("channel/0/item/description",$content);
$xml->save("test.xml","xml");

// ...
// Contentpapst-Framework
// ...
?>
```

6.18.2. Beispiel 2 (RSS-/RDF-Import)

Um externe Inhalte im RSS-/RDF-Format auf der Webseite anzuzeigen, muss stets der Pfad zu einer Vorlage angegeben werden. In dieser Vorlage kann eine freie Formatierung der einzelnen Elemente vorgenommen werden. Es muss hier jedoch stets die Auflistung der Einträge mittels {cp:repeat} intakt bleiben:

```
<strong>{cp:xml:title}</strong><br/>
Beschreibung: {cp:xml:description}<br/>
Webseite: {cp:xml:link}<br/><br/>

{cp:repeat}
- <a href="{cp:xml:link}">{cp:xml:title}</a><br/>
{/cp:repeat}
```

6.18.3. add_attribute()

Jeder Knoten einer XML-Datei kann zusätzlich zum eigentlichen Inhalt (d.h. weiteren Knoten oder einem String) weitere Informationen in Form von Attributen transportieren. Für jedes Attribut muss neben dem Namen und Inhalt des Attributes zusätzlich der Pfad zum gewünschten Knoten angegeben werden.

```
$xml->add_attribute("Shop/Produkte/Fussball","katnummer","1"); // fügt dem Knoten das Attribut katnummer="1" hinzu
```

6.18.4. add_content()

Die Methode "add_content()" fügt einem Knoten weitere Daten hinzu. Sind bereits Daten vorhanden, so werden die neuen Daten angehängt. Der Funktion der Pfad zum genutzten Knoten und der neue Inhalt übergeben.

```
$xml->add_content("Katalog/Produkt/Contentpapst","CMS für den Mittelstand");
```

6.18.5. add_node()

Sofern bereits eingelesene XML-Dateien als Array vorliegen, kann man diesen mit "add_node()" neue Knoten hinzufügen. Hierzu übergibt man der Method den Pfad zum neuen Knoten und den Namen des Knotens. D.h. soll ein Knoten auf der Ebene "Katalog > Produkt" angelegt werden mit dem Namen "Beschreibung", so wendet man die Funktion folgendermaßen an:

```
$xml->add_node("Katalog/Produkt","Beschreibung");
```

6.18.6. convert()

Wenn keine bestehende XML-Datei bearbeitet oder ein externer Webservice genutzt werden soll, kann über die Methode "create" eine neue XML-Datei angelegt werden. Diese Methode muss dann vor allen weiteren Methoden (wie z.B. "add_node") aufgerufen werden:

```
$xml->create();
```

6.18.7. db_input()

"db_input()" ermöglicht die Integration von XML-Daten in eine Datenbank. Dies kann z.B. genutzt werden, um mittels "db_output()" als XML exportierte Datensätze wieder in die Datenbank einzuspielen. Die Methode erwartet als Parameter den Pfad zur XML-Datei, den Tabellennamen (in diese werden die Datensätze eingefügt) sowie den Pfad zum genutzten Knoten.

Alle notwendigen Umwandlungsprozesse werden automatisch vorgenommen. Zusätzlich wird geprüft, ob die Felder in der XML-Datei mit denen der Datenbank übereinstimmen (u.a. Reihenfolge, Feldnamen). Wenn dieser Vorgang erfolgreich abgeschlossen wurde, dann wird TRUE zurückgegeben, andernfalls FALSE.

6.18.8. db_output()

Diese Methode exportiert eine Tabelle einer Datenbank in ein XML-Format und schreibt den Inhalt in eine angegebene Datei. Optional kann als dritter Parameter der Name des Wurzelelements benannt werden.

```
$xml->db_output("test.xml","information"); // legt die Tabelle "information" als XML-Datei "test.xml" an
```

6.18.9. open()

Die Methode "open()" lädt eine XML-Datei aus einem angegebenen Ordner und wandelt den Inhalt komplett in ein Array um. Dieses Array wird für die spätere Verarbeitung benötigt. Über den optionalen Parameter "cache_time" kann angegeben werden, wie lange eine Zwischenspeicherung im Cache vorgenommen werden soll. Diese Zeit wird in Sekunden festgelegt. Wird der Parameter weggelassen, dann werden 3600 Sekunden als Standard verwendet.

```
$xml->open("test.xml"); // lädt die Datei "test.xml"
```

6.18.10. remove_attribute()

Attribute zu Knoten, welche per XML-API angelegt wurden, müssen auch wieder entfernt werden können. Für diese Aufgabe gibt es die Methode "remove_attribute()". Es muss hier lediglich der Pfad zum Knoten angegeben werden.

```
$xml->remove_attibute("Shop/Produkte/Fussball"); // entfernt die Attribute des Knotens
```

6.18.11. remove_content()

Ähnlich wie die Methode "remove_node()" entfernt auch "remove_content()" Bestandteile aus einem Array. In diesem Fall jedoch wird nur der Inhalt eines Knotens gelöscht. Der Knoten selbst bleibt bestehen und kann weiterhin genutzt werden.

```
$xml->remove_content("Katalog/Produkte/Beschreibung"); // "Beschreibung" mit dem Pfad "Katalog/Produkte" wird geleert.
```

6.18.12. remove_node()

Vorhandene Knoten in einem Array können mit dieser Methode entfernt werden. Hierzu übergibt man der Methode lediglich den Pfad zum Knoten. D.h. soll ein Knoten auf der Ebene "Katalog > Produkt" entfernt werden mit dem Namen "Beschreibung", so wendet man die Funktion folgendermaßen an:

```
$xml->remove_node("Katalog/Produkt/Beschreibung");
```

6.18.13. save()

"save()" sorgt für die Aufbereitung der in einem Array vorliegenden Daten in das gewünschte Zielformat. Aktuell werden XML sowie PHP unterstützt. In der XML-Variante wird das komplette Array umgewandelt, die PHP-Variante stellt eine Abbildung des Array-Inhaltes in einer Datei in lesbarer Form dar.

Ein vierter (optionaler) Parameter kann in Form eines Arrays Angaben zur verwendeten Version ("version") und dem Zeichensatz ("encoding") der XML-Datei beinhalten. Weiterhin kann festgelegt werden, ob der generierte Inhalt zurückgeliefert werden soll (in diesem Fall keine Datei angeben), statt ihn in der Datei zu hinterlegen.

```
$xml->save("test.xml","xml",$array,$options); // das Array "$array" wird als XML-Datei "test.xml" exportiert
$xml->save("test.php","php",$array); // das Array "$array" wird als PHP-Datei "test.php" exportiert
```

6.18.14. set_content()

Die Methode "set_content()" füllt einen Knoten mit beliebigen Daten. Sind bereits Daten vorhanden, so werden diese mit den neuen Daten überschrieben. Der Methode werden ein Pfad zum genutzten Knoten und der neue Inhalt übergeben.

```
$xml->set_content("Katalog/Produkt/Contentpapst","CMS für den Mittelstand");
```

6.18.15. show()

Mit der Methode "show()" wird eine verarbeitete Datei im XML-Format zurückgeliefert. Optional kann der einzige Parameter auch auf "0" gesetzt werden. In diesem Fall erfolgt eine direkte Ausgabe an den Browser. Je nach Browser erscheint dann eine Darstellung der XML-Datei in einer Baumstruktur bzw. eine Downloadaufforderung.

```
$xml->show("0"); // XML-Datei wird an den Browser ausgegeben
```

6.18.16. show_feed()

Externe Inhalte im RDF-/RSS-Format können auch per API direkt in eigene Module integriert werden. Die Methode benötigt Parameter zum Pfad der Datei, zum Pfad der verwendeten Vorlage und zur maximalen Anzahl der Einträge. Ein letzter Parameter bestimmt, ob die Inhalte direkt ausgegeben werden ("0") oder zurückgeliefert werden ("1" bzw. weglassen). Alle Dateien werden automatisch zwischengespeichert. Die Ladezeit wird dadurch erheblich reduziert.

```
$xml->show_feed("slashdot.rdf","themes/$theme/xml.htm",5);
// gibt die Inhalte der Datei "slashdot.rdf" mit der Vorlage "themes/$theme/xml.htm" aus (maximal 5 Einträge)
```

6.19. Zeichen & Texte

Diese Funktionen dienen der Manipulation von Zeichen & Texten ("strings") und ermöglichen beispielsweise eine temporäre Verschlüsselung. Um die Funktionen in Module einzubinden, muss die Datei "/functions/cp.string.php" per "include()" oder "include_once()" eingebunden werden.

6.19.1. cp_string_calculate()

Diese Funktion verschlüsselt einen Text standardmäßig über den MD5-Algorithmus. Das Ergebnis kann nicht mehr entschlüsselt werden. Sofern eine PHP-Version > 4.3.0 eingesetzt wird und in den Einstellungen SHA1 als Standard-Verschlüsselung eingestellt wurde, nutzt diese Funktion automatisch den SHA1-Algorithmus.

```
$var = cp_string_calculate($text); // liefert "$text" in der verschlüsselten Fassung zurück
```

6.19.2. cp_string_convert()

Die Funktion "cp_string_convert()" konvertiert ein eindimensionales Array zu einem String. Die einzelnen Array-Elemente werden dabei anhand eines Trennzeichens - standardmäßig "," - miteinander verbunden.

```
$var = cp_string_convert($my_array," - "); // die Elemente des Arrays "$my_array" werden als String zurückgegeben
```

6.19.3. cp_string_convert_unicode()

Mit dieser Funktion kann ein Text vom Unicode-Zeichensatz in einen alternativen Zeichensatz konvertiert werden. Derzeit wird nur der Zeichensatz "ISO-8859-1" unterstützt. Der Funktion muss dazu nur der Ausgangstext übergeben werden:

```
$var = cp_string_convert_unicode($text); // liefert "$text" im neuen Zeichensatz zurück
```

6.19.4. cp_string_count_words()

Diese Funktion zählt die Anzahl der Wörter in einem übergebenen String und liefert diese zurück. U.a. kann "cp_string_count_words()" Anwendung finden in Formularen, in denen die Eingaben durch den Nutzer auf eine bestimmte Wortzahl beschränkt werden sollen.

```
$var = cp_string_count_words($string); // liefert die Anzahl der Wörter in "$string"
```

6.19.5. cp_string_decode()

"cp_string_decode()" entschlüsselt einen beliebig langen String, welcher vorher mit "cp_string_encode()" verschlüsselt wurde. Der Rückgabewert kann dann wieder in eine Variable geschrieben werden.

```
$var = cp_string_decode($string);// entschlüsselt die Variable "$string"
```

6.19.6. cp_string_encode()

"cp_string_encode()" verschlüsselt einen beliebig langen String, welcher dieser Funktion vorher übergeben wurde. Der Rückgabewert kann dann wieder in eine Variable geschrieben werden.

```
$var = cp_string_encode("Hallo"); // verschlüsselt "Hallo"
```

6.19.7. cp_string_rm_spaces()

Diese Funktion wandelt Sonderzeichen um und entfernt Leerzeichen aus einem übergebenen Text.

```
$var = cp_string_rm_spaces($content); // liefert den neuen Inhalt von "$content"
```

6.19.8. cp_string_shorten()

Ähnlich wie die Funktion "cp_word_shorten()" verkürzt "cp_string_shorten()" einen String. In diesem Fall jedoch nicht die einzelnen Elemente, sondern den gesamten String. Hierzu wird eine gewünschte Länge angegeben, standardmäßig sind dies 50 Zeichen. Ein längerer Text wird auf n-3 Zeichen gekürzt und um die Zeichen "..." am Ende ergänzt.

```
$var = cp_string_shorten($text,50); // verkürzt "$text" auf 50 Zeichen
```

6.19.9. cp_string_summarize()

Die Funktion "cp_string_summarize()" verkürzt im Gegensatz zu "cp_string_shorten()" einen Text nicht auf eine maximale Anzahl an Zeichen, sondern führt eine Wortlimitierung durch. Neben dem Text wird hierzu die gewünschte Anzahl an Worten (vom Anfang ausgehend) angegeben. Als Standardwert werden 10 Worte verwendet.

```
$var = cp_string_summarize($text,"20 "); // der Text wird auf 20 Worte gekürzt und zurückgegeben
```

6.19.10. cp_string_swap()

Die Funktion "cp_string_swap()" tauscht den Inhalt zweier Variablen aus. D.h. wenn diese Funktion auf die Variablen $a und $b angewendet wird, dann besitzt anschließend die Variable $a den Inhalt von $b und $b den Inhalt von $a.

```
cp_string_swap($a, $b); // tauscht $a mit $b aus
```

6.19.11. cp_string_unicode()

"cp_string_unicode()" wandelt einen ISO-komformen String in Unicode um. Unicode ist eine Zeichenkodierung, mittels derer sich beliebige Sprachen mit deren jeweiligen Sonderzeichen darstellen lassen. Wollen Sie also sichergehen, dass Texte bei jeder Spracheinstellung gleich aussehen, setzen Sie diese Funktion ein. U.a. lassen sich hiermit auch Email-Adressen vor Spidern schützen, da Unicode für diese nicht lesbar ist.

```
$var = cp_string_unicode($string); // liefert die Unicode-Version von "$string" zurück
```

6.20. ZIP

Mit der ZIP-Klasse werden im ZIP-Format gepackte Dateien gelesen und entpackt. Über weitere Methoden können neue ZIP-Dateien mit beliebigen Inhalten und Ordnern erstellt werden. Um diese Methoden in Module einzubinden, muss die Datei "/functions/cp.zip.php" per "include()" oder "include_once()" eingebunden werden.

6.20.1. Beispiel 1

In diesem Beispiel wird eine einzelne Datei aus einer ZIP-Datei extrahiert und getrennt von dieser gespeichert. Die ZIP-Datei wird durch "open()" zuerst geöffnet und analysiert, "unpack()" führt die Extrahierung durch und "close()" schließt die ZIP-Datei.

```
<?
// ...
// Contentpapst-Framework
// ...

include("functions/cp.zip.php"); // Einbindung der Klasse

$zip = new contentpapst_zip();
$zip->open("sport.zip");
$zip->unpack("bayern_muenchen.jpg","bayern_muenchen.jpg","bilder");
$zip->close();
?>
```

6.20.2. Beispiel 2

Jetzt wird eine neue ZIP-Datei erstellt. In dieser Datei sollen Test-Berichte gespeichert werden. Eine Datei "readme.txt" im Hauptverzeichnis gibt allgemeine Hinweise. Die Testberichte selbst sind jeweils thematisch geordnet in Unterordnern abgelegt. Zuletzt wird die Datei als "testberichte.zip" gespeichert.

```
<?
// ...
// Contentpapst-Framework
// ...

include("functions/cp.zip.php"); // Einbindung der Klasse

$zip = new contentpapst_zip();
$zip->add_dir("musik");
$zip->add_dir("sport");
$zip->add_file("cdplayer.doc","test.cdplayer.doc","musik");
$zip->add_file("fussball.doc","test.fussball.doc","sport");
$zip->add_file("readme.txt");
$zip->pack("testberichte.zip");
$zip->close();
?>
```

6.20.3. add_dir()

Auch Ordner können in der ZIP-Datei erstellt werden. Über die Methode "add_dir()" wird der aktuellen Datei ein neuer Ordner hinzugefügt. Unterordner in der Form "ordner/unterordner" sind dabei ebenfalls möglich.

```
$zip->add_dir("sport/formel1"); // erstellt einen Ordner "sport/formel1" in der ZIP-Datei
```

6.20.4. add_file()

"add_file()" ist die wichtigste Methode für die Erstellung neuer ZIP-Dateien. Über diese Methode bestimmen Sie, welche Dateien in die Datei eingefügt werden. Als Parameter werden der bisherige Dateiname, sowie der gewünschte Name und das Verzeichnis in der ZIP-Datei angegeben (letztere sind optional). Zwingend erforderlich ist allein die Angabe des Pfades.

```
$zip->add_file("ferrari.jpg","mschumacher_ferrari.jpg","sport/formel1");
// hier wird die Datei "ferrari.jpg" als "mschumacher_ferrari.jpg" im passenden Verzeichnis eingefügt
```

6.20.5. close()

"close()" schließt eine geöffnete Datei. Diese Methode muss nur aufgerufen werden, wenn eine vorhandene ZIP-Datei genutzt wird. Sie ist aufzurufen, sobald alle gewünschten Dateien entpackt wurden.

```
$zip->close(); // schließt die aktuelle Datei
```

6.20.6. download()

Die Methode "download()" führt einen direkten Download der zusammengestellten ZIP-Datei durch. Die Methode wird nur intern aufgerufen und benötigt als Parameter den Dateinamen sowie den Inhalt der ZIP-Datei. Je nach Browser werden unterschiedliche Einstellungen vorgenommen. Im Ergebnis sieht der Benutzer stets ein neues Fenster mit einer Download-Aufforderung.

```
$zip->download($file, $content); // Download von "$content" als "$file"
```

6.20.7. open()

Mittels "open()" wird eine vorhandene ZIP-Datei eingelesen. Die enthaltenen Informationen werden überprüft und strukturiert in Form eines Arrays für die weitere Verarbeitung (z.B. über die Methode "unpack()") zur Verfügung gestellt.

```
$zip->open("sport.zip"); // öffnet die Datei "sport.zip"
```

6.20.8. pack()

Die Methode "pack()" dient der Erstellung einer neuen ZIP-Datei. Vorher müssen mittels "add_dir()" neue Ordner (optional) und mittels "add_file()" neue Dateien (notwendig) hinzugefügt wurden sein. "pack()" erhält als Parameter den Dateinamen, eine Angabe, ob eine bereits vorhandene Datei mit dem gleichen Namen überschrieben werden soll (standardmäßig "1") und ob ein automatischer Download durchgeführt werden soll (standardmäßig "0"). Wenn kein Download gewünscht ist, wird die Datei am angegebenen Ort gespeichert.

```
$zip->pack("sport.zip", "1"); // speichert und überschreibt die Datei "sport.zip"
$zip->pack("sport.zip", "0", "1"); // lädt die erstellte ZIP-Datei direkt herunter
```

6.20.9. show_content()

"show_content()" liefert im Gegensatz zu "show_info" ein Array mit Informationen über die einzelnen Dateien in einer ZIP-Datei zurück. Vorraussetzung ist, dass die ZIP-Datei vorher mittels "open()" geöffnet und analysiert wurde. Die Informationen beinhalten u.a. Größe, Erstellungsdatum und Typ der Datei sowie einen optionalen Kommentar. "show_content()" dient der externen Weiterverarbeitung der Dateiinformationen. Über einen optionalen Parameter können gezielt Informationen über eine Datei herausgegriffen werden.

```
$zip->show_content(); // liefert Informationen über alle enthaltenen Dateien
$zip->show_content("readme.txt"); // liefert Informationen über die Datei "readme.txt"
```

6.20.10. show_info()

"show_info()" liefert ein Array mit Informationen über die Struktur der ZIP-Datei zurück. Vorraussetzung ist, dass die ZIP-Datei vorher mittels "open()" geöffnet und analysiert wurde. Die Informationen beinhalten u.a. die Anzahl der enthaltenen Dateien sowie die Größe und einen optionalen Kommentar. "show_info()" wird intern genutzt, die Informationen sind jedoch auch für die externe Weiterverarbeitung abrufbar.

```
$zip->show_info(); // liefert Informationen über die aktuelle Datei
```

6.20.11. unpack()

"unpack()" entpackt eine Datei aus einer ZIP-Datei und speichert diese unter einem neuen Namen im angegebenen Verzeichnis. Hierzu werden als Parameter der ursprüngliche Name sowie der neue Name der Datei und ein Verzeichnis benötigt. Im Erfolgs- oder Misserfolgsfall werden Statusmeldungen zurückgegeben (siehe "Statusmeldungen").

```
$zip->unpack("alte_datei.txt","neue_datei.txt","pfad/verzeichnis"); // entpackt die Datei "alte_datei.txt"
```

6.20.12. Statusmeldungen

Je nach genutzter Methode werden unterschiedliche Statusmeldungen über Erfolg/Misserfolg einer Aktion zurückgegeben. Im Folgenden finden Sie eine Liste der häufigsten Meldungen:

0	Leeres Ergebnis, Information/Datei ist nicht vorhanden.
1	Vorgang erfolgreich abgeschlossen.
2	Die Zieldatei konnte nicht geschrieben werden.
3	Es wurde keine Zieldatei angegeben. (Pack)
4	Cache-Datei konnte nicht geöffnet werden.
5	Datei konnte nicht im Archiv gefunden werden. (Unpack)
6	Benötigte Erweiterung "zlib" und die Funktion "gzopen" konnten nicht geladen werden. (Unpack)
7	Die Zieldatei ist bereits vorhanden.
8	Es wurden bisher keine Dateien hinzugefügt.

7. FAQ - Häufig gestellte Fragen

Hier haben wir Antworten auf Fragen zusammengetragen, welche uns von unseren Kunden häufig gestellt werden. Diese Fragen können sowohl technischer, als auch gestalterischer oder inhaltlicher Natur sein. Sollten Sie eine weitere Frage zu diesem System haben, so kontaktieren Sie uns über den Kundenservice. Mit jedem Update des Systems wird die FAQ dann um weitere hilfreiche Antworten für die wichtigsten Fragen ergänzt.

7.1. Übersicht

7.1.0. Bei der Nutzung des Internet Information Server (IIS) treten vereinzelt Probleme auf!?

Wenn Sie den Internet Information Server (IIS) von Microsoft einsetzen, dann werden intern einige Systemvariablen von PHP anders verwaltet und geben gegenüber dem gewöhnlich genutzten Apache-Webserver andere Werte zurück. U.a. betroffen ist davon die Variable "$REQUEST_URI". Diese muss für den IIS in "$PATH_INFO" geändert werden.

Da die Variable jedoch in verschiedenen Dateien des System eingesetzt wird, empfiehlt sich die Bearbeitung über einen Text-Editor, mit welchem Sie den kompletten Ordner durchsuchen und automatisch alle Vorkommen der Variablen in "$PATH_INFO" ändern können.

7.1.1. Beim Anlegen neuer Sprachen fehlen einige Ordner! Was ist los?

Wenn Sie eine neue Sprache anlegen, d.h. eine Sprache auswählen und die passenden Flaggen dazu hochladen, dann wird **Contentpapst** ordnungsgemäß die Sprachdateien der aktuell eingestellten Sprache duplizieren und in neuen Ordnern ablegen. Je nach Einstellung des Webservers besitzt ein PHP-Script aber nicht immer das Recht, neue Ordner oder Dateien anzulegen. Sie müssen in diesem Fall auf die "languages"-Ordner der einzelnen Module und gegebenenfalls ebenso auf den Ordner "/languages/" im Hauptverzeichnis per FTP einen CHMOD von 755 bzw. 777 setzen (weitere Informationen zum CHMOD finden Sie in der Installationsanleitung). Dann darf auch **Contentpapst** neue Ordner anlegen und es können alle für die neue Sprache erforderlichen Dateien angelegt werden.

7.1.2. Es treten beim Absenden von Formularen Fehler auf. Es wird immer ein Download-Fenster geöffnet!

Contentpapst nutzt für die schnellere Auslieferung der Seiteninhalte eine serverseitige Komprimierung / Verschlüsselung auf der Basis des GZIP-Algorithmus. Es wird vom System automatisch geprüft, ob dieser von Ihrem Webserver / Browser unterstützt wird. Je nachdem ob dies der Fall ist, wird die GZIP-Komprimierung aktiviert oder nicht. Allerdings ist in einigen Fällen eine automatische Überprüfung von Webserver und (seltener) Browser nicht möglich.

U.a. zeigt der Microsoft Internet Explorer bei einigen Versionen > 4.0 teilweise ein fehlerhaftes Verhalten. Dieser fordert Inhalte per HTTP 1.0 an. Die Entschlüsselung von GZIP-Inhalten koppelt er jedoch an HTTP 1.1. Erhält der Internet Explorer dann GZIP-Inhalte als HTTP 1.0 (wie dies standardmäßig vom Webserver geliefert wird), ignoriert er den angegebenen "Content-Type" (GZIP). Da ein passender "Content-Type" nicht feststellbar zu sein scheint, erklärt dieser Browser die GZIP-Inhalte als Download und zeigt diese nicht direkt im Browserfenster an. Alle anderen Browser reagieren hier wie in den Standards vorgegeben.

Diesem Verhalten des Internet Explorers kann manuell Abhilfe geschaffen werden, indem in den Einstellungen des Browsers unter "Extras > Internetoptionen > Erweitert" die Optionen "HTTP 1.1 verwenden" und "HTTP 1.1 über Proxyverbindungen verwenden" aktiviert werden.

7.1.3. Ich habe die Verschlüsselung per SHA1-Algorithmus aktiviert, jetzt treten Probleme auf!

Wenn Sie die Verschlüsselung vom MD5- auf den SHA1-Algorithmus umgestellt haben und es zu Problemen kommt (beispielsweise da die Passwörter der einzelnen Benutzer noch in MD5-Verschlüsselung in der Datenbank liegen), müssen Sie dies manuell rückgängig machen. Öffnen Sie hierzu per FTP die Datei "config.db.php" in Ihrem Vorlagenset. Dort finden Sie diese Zeile:

```
$configuration["main"]["sha1_support"] = "1";
```

Deaktivieren Sie SHA1, in dem Sie die Zeile abändern auf:

```
$configuration["main"]["sha1_support"] = "0";
```

Anschließend loggen Sie sich als Administrator ein und setzen Sie die SHA1-Verschlüsselung über "Einstellungen > Allgemeine Einstellungen" wieder zurück.

7.1.4. Ich habe mein Passwort verloren, wie kann ich es zurückerhalten?

In diesem Fall gibt es zwei Möglichkeiten. Da das Passwort verschlüsselt ist, kann ihr originales Passwort nicht mehr zurückgewonnen werden. Der Benutzer-Bereich bietet Ihnen jedoch eine Passwort-Recovery-Funktion an (beim Klick auf "Passwort vergessen?"), welche ein neues Passwort generiert. Geben Sie hierfür Ihren Benutzernamen und die Email-Adresse an, mit welcher Sie sich registriert haben. Ein neues Passwort wird Ihnen per Email mitgeteilt.

Alternativ dazu können Sie - sofern Sie vollen Datenbankzugriff besitzen - mit einem entsprechenden Tool (z.B. phpMyAdmin) in der Datenbanktabelle "*_user" ("*" ist durch den bei der Installation gewählten Datenbankpräfix zu ersetzen) Ihren Benutzer-Eintrag ändern. Suchen Sie Ihren Benutzernamen und ändern Sie das Passwort auf "21232f297a57a5a743894a0e4a801fc3".

Loggen Sie sich danach im Benutzer-Bereich mit Ihrem Benutzernamen und dem Passwort "admin" ein. Bitte ändern Sie nach Möglichkeit dieses Passwort sofort nach dem Login in Ihren Einstellungen.

7.1.5. Ist Contentpapst XML-kompatibel?

Contentpapst beinhaltet umfangreiche Funktionen für den Import und Export von XML-Inhalten. Diese können genutzt werden, um XML-Inhalte aus dem Internet live auf der Webseite in einem frei definierbaren Layout anzuzeigen (beispielsweise RSS-Feeds). Der Export von Daten im XML-Format ermöglicht zudem die Interaktion mit externen System (u.a. Warenwirtschaftssystem und CRM). Weiterhin enthält das System eine komplette Webservices-Schnittstelle, die neben dem Export in XML über XSLT-Stylesheets den Export in fast beliebigen Formaten ermöglicht. Weitere Informationen zur Webservices-Unterstützung bietet der Abschnitt "Systemkomponenten / Module > Webservices".

7.1.6. Ist die Installation von Software / PlugIns auf dem lokalen Rechner notwendig?

Dies ist nicht notwendig. **Contentpapst** wird allein über den Browser bedient, wofür Sie eine möglichst aktuelle Browserversion verwenden sollten. PlugIns sind nur notwendig, wenn die zu erstellende Webseite solche voraussetzt (z.B. das Flash-PlugIn von Adobe). Sie entscheiden also selbst, wie Sie die Vorlagen Ihrer Webseite gestalten und mit welchen Technologien Sie diese ausstatten. Von der suchmaschinenfreundlichen und behindertengerechten Portalseite bis hin zu einer aufwändigen, mit Flash angereicherten Unternehmenspräsentation ist alles möglich.

7.1.7. Ist die Verwaltung großer Dateien (> 1 MB) möglich?

Contentpapst greift bei der Dateiverwaltung auf die von Ihrem Webhoster für die Datenbank bzw. Ihren Webspace gemachten Einschränkungen zurück. Allgemein wird der Upload ab einer bestimmten Dateigröße (z.B. 4-8 MB) begrenzt. Auch Datenbanken sind in ihrer Größe oft limitiert (jedoch meist erst ab mehreren Hundert MB). Informieren Sie sich bei Ihrem Webhoster, wie dessen Einschränkungen aussehen. **Contentpapst** wird sich daran orientieren.

7.1.8. Ist ein Einsatz im Intranet oder Extranet möglich?

Contentpapst kann auch im Intranet oder Extranet von Unternehmen oder Organisationen eingerichtet werden. Wichtig ist jedoch, dass auch hier die allgemeinen Systemvoraussetzungen erfüllt werden. Informieren Sie sich auch über zusätzlich angebotene Module aus den Bereichen HR und CRM, welche sich für Installationen im Intranet besonders gut eignen.

7.1.9. Kann ich meine Seite gezielt für Suchmaschinen optimieren?

Ja, **Contentpapst** bietet Ihnen standardmäßig eine Reihe von Mitteln zur Optimierung der Seite für Suchmaschinen an. Die wichtigsten seien hier kurz genannt:

- Zuallererst sollten Sie Ihre Vorlagen auf Validität überprüfen lassen durch den Validator des W3C, denn korrekt geschriebene Seiten können von Suchmaschinen leichter erfasst und verarbeitet werden.
- Für jeden Artikel sollten Sie die Metatags (unter "Artikel bearbeiten") möglichst präzise ausfüllen, auch die allgemeinen Metatags (unter "Einstellungen > Metatags") sind bedeutsam (vor allem Titel und Schlüsselwörter).
- Wenn Ihr Server "mod_rewrite" unterstützt, sollten Sie die Suchmaschinenfreundlichen URLs unter "Einstellungen > Suchmaschinenfreundliche URLs" aktivieren.
- Zusätzlich können Sie für Artikel, Artikelkategorien, News und viele weitere Bereiche über die "Permalinks" ganz individuelle URL-Strukturen nutzen, die sich sehr gut zur Optimierung der Webseite auf bestimmte Begriffe ("Keywords") eignen.

Da das Aufgabengebiet der Suchmaschinenoptimierung mittlerweile sehr umfangreich geworden ist, können wir Ihnen hier nur einige Tipps geben. Doch wenn Sie allein diese Tipps einhalten, wird Ihre Seite bereits erheblich suchmaschinenfreundlicher werden und kann somit - einige Zeit nachdem die Suchmaschinen-Spider die Webseite besucht und in den Index aufgenommen haben - in den Suchmaschinen besser gefunden werden.

Tipp: Testen Sie mit dem "Poodle Predictor", wie Ihre Seite später in den Suchergebnissen von Google ausschauen wird!

Die sandoba.de medien agentur bietet selbst Suchmaschinenoptimierungen für Seiten an, die mit **Contentpapst** oder auch statisch umgesetzt wurden. Diese Dienstleistungen umfassen u.a. die Optimierung der Webseite für bestimmte Begriffe ("Keywords"), die Anpassung des Quellcodes der Vorlagen, eine regelmäßige Erfolgskontrolle usw.

7.1.10. Können dynamische Inhalte anderer Anbieter importiert werden?

Ja, dies ist möglich, allerdings abhängig von der Art der Inhalte. XML-Inhalte können Sie ebenso direkt importieren und anzeigen, wie z.B. Inhalte per Javascript. Ein Import von CSV- oder SQL-Inhalten kann im Administrationsbereich (im Modul "Sicherung / Datenbank") vorgenommen werden. Für andere Inhalte und Scripte (PHP usw.) gibt es die Möglichkeit, diese direkt in einer Vorlage einzubinden (d.h. die Scripte in einem Artikel oder einer Vorlage zu platzieren) bzw. diese über den Tag {cp:include} durch **Contentpapst** verarbeiten zu lassen.

7.1.11. Können {cp:tags} in PHP-Dateien verarbeitet werden?

Standardmäßig werden {cp:tags} die in der Ausgabe einer über {cp:include} eingebundenen PHP-Datei enthalten sind aus Performancegründen nicht verarbeitet. Dies können Sie jedoch leicht manuell aktivieren. öffnen Sie hierzu die Datei "/functions.php" und fügen Sie in der Funktion "cp_include" vor "RETURN $string;" folgenden Quellcode ein:

```
include_once("functions/cp.template.php");
$string = cp_template_output(cp_template_process($string), "1", "0");
```

Dann wird vor der endgültigen Ausgabe der angelieferte Inhalt überprüft, alle darin enthaltenen {cp:tags} werden verarbeitet und deren Rückgaben werden anstelle des {cp:tags} integriert.

7.1.12. Lassen sich bestehende Inhalte in Contentpapst übernehmen?

Die Übernahme bestehender Webseiten ist mit **Contentpapst** keineswegs mit viel Arbeit verbunden. Richten Sie hierfür zuerst ein neues Vorlagen-Set ein, in welches Sie den vorhandenen HTML-Code integrieren (nur das Layout ohne Text-Inhalte usw.). Wichtig sind hierbei Vorlagen für die gesamte Seite sowie die Aufteilung der Inhalte in Tabellen und einzelne Unterseiten.

Laden Sie dann die Bilder und alle anderen benötigten Dateien in die jeweiligen Ordner mittels der Dateiverwaltung oder per FTP hoch und treffen Sie die gewünschten Einstellungen am System. Nun können Sie mit dem Einfügen der Inhalte als News, Artikel oder einzelne Dateien beginnen. Speziell Webseiten mit einer klaren Gliederung können auf diesem Wege schnell in **Contentpapst** übernommen werden.

7.1.13. Nach dem Ändern von Einstellungen werden diese nicht aktualisiert. Warum?

Dies deutet darauf hin, dass der CHMOD bei der Installation nicht korrekt gesetzt wurde. Insbesondere werden für den Hauptordner mit der Datei "config.php" und die Datei "config.db.php" in dem Vorlagenset des Mandanten ein CHMOD von 755 oder 777 (je nach Webspace) benötigt. In der Datei "config.db.php" werden nach Änderungen im Administrationsbereich alle Einstellungen zwischengespeichert, um die erforderlichen Datenbankabfragen weitest möglich zu begrenzen.

Ist eine Aktualisierung dieser Dateien durch das System nicht möglich, wird zwar die Datenbank aktualisiert, auf der Webseite sind jedoch weiterhin die alten Einstellungen aktiv. Ändern Sie daher den CHMOD für die Dateien und Ordner, wie dies in der Installationsanleitung des Systems beschrieben wird.

7.1.14. Nach der Aktivierung der Suchmaschinenfreundlichen URL's funktioniert die Seite nicht mehr!

Wenn unter "Einstellungen > Suchmaschinenfreundliche URL's " die Suchmaschinenfreundlichen URL's aktiviert werden, dann sollten die "mod_rewrite"-Regeln automatisch in die Datei ".htaccess" geschrieben werden. Sollte dies nicht automatisch vorgenommen werden, dann erhöhen Sie bitte per FTP kurzfristig den CHMOD der Datei ".htaccess" auf 755 bzw. 777. Dann deaktivieren und aktivieren Sie die Suchmaschinenfreundlichen URL's wieder über den Administrationsbereich. Anschließend sollten die mod_rewrite-Regeln in der Datei ".htaccess" vorhanden sein. Nun kann der CHMOD per FTP bei Bedarf wieder auf den vorherigen Stand zurückgesetzt werden.

Noch ein Hinweis: Wenn das System in einem Unterordner betrieben wird, dann ist vor der Aktivierung unter "Einstellungen > Allgemeine Einstellungen" noch der Pfad zum CMS einzugeben. Nur dann können über "mod_rewrite" die dynamischen URL's korrekt angesteuert werden.

7.1.15. Sind fertige Vorlagen-Sets verfügbar?

Jeder Lizenz von **Contentpapst** liegt mindestens ein fertiges Vorlagen-Set bei. Die Vorlagen-Set dienen als Demonstration der Funktionen von **Contentpapst** und zum besseren Verständnis der Arbeitsweise. Weitere Vorlagen-Sets können Sie bei uns oder kooperierenden Agenturen bestellen. Die Vorlagen-Sets sind stets in einem Unterordner des Ordners "/themes/" ausgehend vom **Contentpapst**-Hauptordner hinterlegt.

7.1.16. Sind zusätzliche Schulungen erforderlich?

Ob ein Schulungsbedarf besteht, ist unterschiedlich von der Art der Nutzung von **Contentpapst**. Redakteure arbeiten komplett innerhalb des Administrationsbereiches im Browser und können **Contentpapst** innerhalb nur weniger Stunden oder Tage verstehen und bedienen. Für Entwickler gibt es zusätzliche Support-Möglichkeiten wie die integrierte Online-Hilfe, Support per Email oder im Forum sowie Tipps&Tricks in der Knowledge Base und dem Entwickler-Blog.

7.1.17. Unterstützt Contentpapst mehrere Sprachen?

Ja, **Contentpapst** unterstützt eine beliebige Anzahl an Sprachen. Diese Sprachen werden über das Modul "Sprachverwaltung" gepflegt. Dort können neue Sprachen hinzugefügt oder vorhandene Sprachen entfernt werden. Zusätzlich bietet das Modul eine schnelle Möglichkeit, einzelne Sprachelemente der Module zu bearbeiten (d.h. die vorhandenen Übersetzungen abzuändern usw.).

Wenn Sie eine neue Sprache hinzugefügt haben, dann können Sie in der oberen rechten Ecke des Administrationsbereiches die neue Sprache im Dropdown-Menü auswählen. Sobald Sie diese auswählen, wechselt die Sprache und Sie können sofort die gewünschten Inhalte in der neuen Sprache eingeben. Auf der Benutzerseite können Links für den Wechsel der Sprache durch den {cp:tag} "{cp:language}" eingeblendet werden.

7.1.18. Warum werden nicht alle Inhalte nach dem Bearbeiten aktualisiert?

Wenn nach dem Bearbeiten von Inhalten (beispielsweise einem Artikel oder einer News) diese nicht sofort auf der Webseite aktualisiert werden, so kann dies an den Einstellungen des Caches von **Contentpapst** liegen. Soll der jeweilige Inhalte dennoch sofort und damit vor Ablauf der Cache-Zeit in der jeweils neuesten Fassung online angezeigt werden, so haben Sie unter "Einstellungen > Cache" die Möglichkeit, den Cache zurückzusetzen.

7.1.19. Welche Aufgabe hat die Datei ".htaccess"?

Die Datei ".htaccess" im Hauptordner beinhaltet einige Einstellungen, die den Webserver in seinem Verhalten beeinflussen. Eine wichtige Rolle spielt diese Datei bei den "Suchmaschinenfreundlichen URL's" (mittels "mod_rewrite"). Weiterhin wird der Zugriff auf Dateien mit den Endungen "cfg", "class", "inc", "mod", "lng" "tmp" und "tpl" eingeschränkt und als allgemeine Fehlerseite wird schließlich noch die Startseite der Webseite festgelegt.

7.1.20. Welche Limitierungen bei der Erstellung des Layouts gibt es?

Dies ist einfach zu beantworten - Keine! Sie sind vollkommen frei bei der Erstellung des Layouts und nicht an bestimmte Schemata gebunden. Das Layout einer Webseite wird aus verschiedenen Vorlagen zusammengesetzt. Diese Vorlagen sind reine HTML-Dateien, erweitert um die {cp:tags}. Somit können Sie wie gewohnt Ihre Webseite erstellen und dabei alle aus HTML bekannten Funktionen nutzen.

Beim Aufbau Ihrer Seite haben Sie im Wesentlichen folgende Möglichkeiten:

- Wenn Sie viele unterschiedliche Layouts verwenden, legen Sie einzelne HTML-Vorlagen mit den gewünschten {cp:tags} an und rufen Sie diese über "index.php?static=[PFAD ZUR DATEI]" auf.
- Sie gehen vom Standard-Vorlagenset aus und erstellen auf dieser Basis mit den mindestens benötigten Vorlagen Ihr eigenes Layout, welches jederzeit erweitert werden kann.
- Sie nutzen XSLT-Stylesheets auf der Basis der Webservices zur Darstellung der Inhalte.

Den gängigsten Weg stellt dabei die zweite Möglichkeit dar. Die beiden anderen sind nur zu empfehlen, wenn Sie bereits über umfangreichere Erfahrungen mit **Contentpapst** verfügen oder eine Webseite mit sehr stark variierenden Darstellungen der Inhalte gestalten müssen.

7.1.21. Wie kann ich die Fehlerseiten ändern?

Contentpapst wechselt bei einer nicht vorhandenen Datei automatisch zurück auf die Startseite der Webseite. Dieser Wechsel wird vollzogen bei folgenden Fehlercodes:

- *403 Forbidden* - der Server verweigert die Ausführung
- *404 Not Found* - der Server kann die Methode nicht ausführen, weil die Seite nicht gefunden wurde
- *500 Internal Server Error* - beim Server gab es einen internen Fehler

In der Datei ".htaccess" im Hauptverzeichnis des Systems wird für jeden dieser Fehlercodes die entsprechende Alternativ-Seite festgelegt. Dort können Sie beispielsweise die URL zu einem Artikel integrieren, in welchem Sie dem Besucher einen Hinweis auf die nicht vorhandene Seite geben und zugleich eine Sitemap oder Suche anbieten.

7.1.22. Wie sieht es mit der Sicherheit aus?

Alle sensitiven Daten der Mitglieder, Redakteure und Administratoren sind durch Benutzername und Passwort gesichert. Das Passwort wird zusätzlich mittels des MD5-Algorithmus verschlüsselt und liegt auch in der Datenbank nicht in entschlüsselter Form vor. Eine für jeden Benutzer spezifisch erzeugte Session erlaubt den Schutz von Daten. Der MD5-Algorithmus ermöglicht zudem höchste Sicherheit, bisher ist eine Entschlüsselung nicht möglich. Alternativ können Sie zur Verschlüsselung auch den SHA1-Algortihmus verwenden (weniger bekannt, Umstellung unter "Einstellungen > Allgemeine Einstellungen"). Für vergessene Passwörter bietet **Contentpapst** eine Passwort-Recovery-Funktion per Email an. Hierfür wird ein neues Passwort generiert.

7.1.23. Wie werden eigene Vorlagen-Sets erstellt?

Die Anpassung des Layouts geht in wenigen Schritten. Kopieren Sie einen der vorhandenen Vorlagen-Ordner und passen Sie die Dateien entweder mit dem integrierten WYSIWYG-Editor oder einem externen Programm an. Dazu gehören auch die Grafiken im Unterordner "/images/". In die HTML-Vorlagen sind die im Handbuch beschriebenen {cp:tags} einzubetten. An diesen Stellen wird später der Inhalt eingesetzt. Beim Aufbau des Layouts sind Sie vollkommen frei.

7.1.24. Wie werden Medien (Bilder, Videos, PDFs) gespeichert?

Alle Dateien, die nicht Artikel/News, d.h. direkter Content, sondern Dokument sind, werden auch als solche physikalisch im Dateisystem gespeichert. Eine Speicherung innerhalb der Datenbank würde keine Vorteile bringen und nur für eine unnötige Belastung sorgen. Erleichtert wird Ihnen die Arbeit mit dem Dateisystem durch die in **Contentpapst** integrierte Dateiverwaltung (im Administrationsbereich unter "Dateien").

7.1.25. Wie kann man temporäre Dateien im Ordner "cache" löschen?

Dies ist generell über den Administrationsbereich unter "Einstellungen & Cache" möglich. Es werden dann automatisch alle Dateien aus diesem Ordner gelöscht, so dass bislang zwischengespeicherte Seitenelemente beim nächsten Aufruf neu generiert werden.

Alternativ zum manuellen Aufruf dieser Funktion kann allerdings auf einem eigenen Server der Cache auch automatisiert über einen Cronjob geleert werden. Ein Cronjob kann dann beispielsweise mit folgendem Code alle Dateien aus dem Cache löschen, die älter als 5 Tage sind:

```
find cache/ -mtime +5 -type f -exec rm -vf '{}' \;
```

7.2. Lizenzbedingungen

7.2.1. Darf Contentpapst zur Entwicklung auf einer anderen Seite installiert werden?

Wenn Sie Contentpapst vor der Fertigstellung Ihres Projektes zuerst testweise auf einer anderen Seite (d.h. unter einer anderen URL) installieren, so müssen Sie sich einmalig registrieren. Sobald die Entwicklung abgeschlossen ist und Contentpapst auf die richtige Seite geladen werden soll, teilen Sie uns dies bitte per Email mit (über den Kundenservice). Die Registrierung Ihrer Lizenz wird dann abgeändert. Die vorherige Contentpapst-Installation muss dabei entsprechend gelöscht werden.

7.2.2. Können mehrere Webseiten mit einer Contentpapst-Lizenz angelegt und verwaltet werden?

Dies ist abhängig von Ihrer Lizenz. Besitzen Sie die Lizenz für die Grundversion von Contentpapst, so it die Verwaltung von mehreren Webseiten nicht möglich. Sie können nur eine Webseite anlegen, welche jedoch über verschiedene Domains erreichbar sein darf. Für Agenturen und Internet Service Provider können alternative Lösungen bereitgestellt werden, welche die zentrale Administration einer größeren Anzahl von Webseiten erlauben. Beachten Sie in diesem Zusammenhang bitte auch die Lizenzbedingungen.

8. Sonstige Informationen

In diesem Bereich sind Informationen zu zusätzlichen Dienstleistungen rund um das System zu finden. Dies umfasst u.a. die Lizenzbedingungen, das Dienstleistungsangebot des Herstellers sowie das Partnerprogramm für die Systeme.

8.1. Übersicht

8.1.0. Lizenzbedingungen

Lizenzbedingungen

Die über die Webseite http://www.sandoba.de/ bzw. in Online-Shops, über Partner-Unternehmen oder in Ladengeschäften erworbene Lizenz dieses Produktes unterliegt nachfolgenden Bedingungen. Mit dem Kauf des Produktes haben Sie den Lizenzbedingungen zugestimmt. Es gelten die gesetzlich allgemein

vorgeschriebenen Bedingungen für die Haftungen an Produkten.

1. Urheberrecht

Die Rechte am erworbenen Produkt liegen ausschließlich bei der sandoba.de medien agentur, Inh. Robert Fischer. Mit dem Kauf wird nur das Nutzungsrecht, nicht jedoch das Urheberrecht am Produkt auf den Käufer übertragen. Diese Rechte sind durch Verträge und allgemein gültige Rechtsvorschriften abgesichert. Es ist Ihnen erlaubt, für die private Sicherung EINE Kopie des Produktes zu erstellen. Weitergabe, Vermietung oder Verleih im Ganzen oder in Teilen sind jedoch ausdrücklich untersagt.

Ein Weiterverkauf einer Lizenz ist sofort bei der sandoba.de medien agentur zu melden, die bisherige Lizenz ist komplett zu deinstallieren, alle Kopien sind zu entfernen und die Kontaktdaten des neuen Besitzers müssen der sandoba.de medien agentur mitgeteilt werden, damit die Lizenz auf diesen übertragen werden kann.

2. Nutzungsrecht

Je nach der von Ihnen erworbenen Lizenz sind eine, zwei ("2er-Pack"), fünf ("5er-Pack"), 10 ("10er-Pack"), 20 ("20er-Pack") oder unlimitierte ("enterprise") Installationen des Produktes auf einem Webserver erlaubt. Auf die Installation des Produktes darf über verschiedene Domains zugegriffen werden, jedoch muss es sich um den logisch gleichen Inhalt handeln. Eine mehrfache Installation einer einfachen Lizenz auf einem Webserver ist untersagt und wird von der sandoba.de medien agentur verfolgt.

Das Nutzungsrecht einer Produkt-Lizenz geht erst auf den Käufer über, wenn dieser ausstehende Forderungen gegenüber der sandoba.de medien agentur vollständig beglichen hat. Ist dies nicht innerhalb eines angemessenen Zeitraumes möglich, erhält die sandoba.de medien agentur das Recht, die Lizenz des Produktes auf Kosten des Käufers zurückzunehmen.

3. Eingriff in den Quellcode

Der Quelltext des Produktes liegt offen, so dass ein Eingriff durch den Käufer in den Quelltext jederzeit möglich ist. Änderungen und Erweiterungen dürfen soweit technisch machbar durchgeführt werden, auch die Anbindung des Produktes an ein anderes System bzw. die Integration fremder Systeme in das Produkt ist erlaubt. Zu beachten sind hierbei jeweils die Lizenzbestimmungen der fremden Systeme. Es dürfen hieraus keine Ansprüche gegenüber der sandoba.de medien agentur entstehen.

Das Urheberrecht - auch an der geänderten Lizenz - liegt weiterhin beim Hersteller. Stets beibehalten werden müssen Hinweise auf den Hersteller im Administrationsbereich (u.a. Copyright-Angabe, Verlinkung zu http://www.sandoba.de/) und im öffentlichen Bereich der für den Nutzer nicht sichtbare Kommentar im Quelltext (Information zum Produkt, Versionsnummer und Hersteller). Separate Vereinbarungen zwischen Premium-Partnern und der sandoba.de medien agentur sind hiervon nicht betroffen.

4. Haftung für das Produkt

Die sandoba.de medien agentur haftet für schwerwiegende Fehler des Produktes mit maximal der Höhe des Kaufpreises. Ausgeschlossen ist dies jedoch bei nicht sachgemäßer Handhabung (bspw. wie in der Anleitung beschrieben). Kleinere Fehler ("Bugs") sind unverzüglich über die bekannten Kontakt-Möglichkeiten der sandoba.de medien agentur mitzuteilen, so dass diese behoben werden können (z.B. im nächsten Produktupdate oder -upgrade).

Es wird ebenfalls keine Haftung übernommen, wenn das Produkt durch Dritte bzw. in Kombination mit fremder Software oder Hardware genutzt wird. Es wird nur die Lauffähigkeit der Software wie in den Systemanforderungen angegeben garantiert. Für die Nutzung des Produktes werden der Firefox ab Version 2.0 bzw. der Internet Explorer ab Version 6.0 von Microsoft empfohlen. Die Nutzung mit anderen Browsern ist möglich, jedoch kann keine Haftung auf fehlerhafte Funktionen (Funktionen, welche in diesem Browser nicht funktionieren) übernommen werden.

Um sich gegen den Verlust von Daten abzusichern, ist ein regelmäßiges Backup ratsam. Die Produkte "**Contentpapst**" und "CP::Shop" bieten hierzu die benötigten Funktionen. Weiterhin ist eine Datensicherung auch über die üblichen externen Massnahmen realisierbar. Wird ein regelmäßiges Backup unterlassen, tritt keine Haftung der sandoba.de medien agentur bei Datenverlust ein - ausgenommen sind schwerwiegende Fehler des Produktes.

5. Updates, Upgrades

Updates für das Produkt können mittels der Seriennummer und der übrigen Kundendaten über das Kunden-Konto auf http://www.sandoba.de/ bezogen werden. Nach dem Kauf eines Produktes hat der Käufer für ein volles Jahr kostenlos Zugriff auf die für das Produkt jeweils verfügbaren Updates. Maßgeblich bedeutend ist das Datum des Kaufes der Lizenz, welches auf der Rechnung vermerkt ist. Nach Ablauf dieses Jahres können Updates und Upgrades zu einem ermäßigten Preis erworben werden. Durch diese Updateverlängerungen erhält der Käufer dann erneut für ein Jahr kostenlosen Zugriff auf die Updates.

6. Erfüllungsort, Gerichtsstand, sonstige Bestimmungen

Erfüllungsort und Gerichtsstand für alle Leistungen von Käufer und Hersteller ist der Sitz der sandoba.de medien agentur. Der Vertrag unterliegt für beide Vertragspartner dem Recht der Bundesrepublik. Sollte eine Bestimmung dieser Lizenzbedingungen ungültig sein oder werden, bleiben die Lizenzbedingungen samt aller übrigen Bestimmungen gültig. Die beanstandete Klausel ist durch eine solche zu ersetzen, die dem Zweck der ursprünglichen möglichst nahe kommt. Entsprechendes gilt für Lücken. Bei einem Verstoß gegen die Lizenzbedingungen kann die Lizenz des erworbenen Produktes jederzeit durch die sandoba.de medien agentur zurückgezogen und für nichtig erklärt werden, Rückzahlungen für erworbene Lizenzen entfallen in diesem Fall.

Haben Sie Fragen zu den Lizenzbestimmungen, so wenden Sie sich bitte an den Support (per Email an contact@sandoba.de oder über unsere Website http://www.sandoba.de/).

8.1.1. Über den Hersteller

Das Content-Management-System **Contentpapst** ist ein Produkt der sandoba.de medien agentur aus Berlin. sandoba.de entwickelt gezielt E-Business- und E-Commerce-Lösungen für kleine und mittelständische Unternehmen. Zu den Kerngebieten zählt die Programmierung von dynamischen Webapplikationen auf PHP/MySQL-Basis, erweitert um Technologien wie XML/XSL, Datenbanksysteme (neben MySQL u.a. PostGreSQL, Oracle, ODBC) usw., sowie die Erstellung von Internetauftritten und sonstigen multimedialen Präsentationen. Die sandoba.de medien agentur ist unter http://www.sandoba.de/ zu erreichen.

8.1.2. Partnerprogramm für Produkte von sandoba.de

Wenn Sie - wie wir es Ihnen wünschen - mit **Contentpapst** zufrieden sein sollten, können Sie uns gerne weiterempfehlen. Und dabei können Sie auch zusätzlich verdienen! Die sandoba.de medien agentur bietet für **Contentpapst** ein Partnerprogramm mit fester Provision am durch Sie erzielten Umsatz durch Käufe des Systems an. Informieren Sie sich unter http://www.sandoba.de/unternehmen/vertriebspartnerschaft/ über das Partnerprogramm und wie Sie Ihre Provisionserlöse durch Werbung steigern können.

8.1.3. Layouterstellung, Installationsservice und weitere Dienstleistungen

Die sandoba.de medien agentur erstellt mit einem erfahrenen Team komplette Vorlagensets für Ihre Webseite ganz nach Ihren Vorgaben. Liegt Ihnen bereits ein fertiges Layout als Grafik vor oder soll das Layout einer bestehenden Webseite in **Contentpapst** eingebaut werden, so können wir Ihnen selbstverständlich ebenfalls im Rahmen unseres Dienstleistungsangebotes weiterhelfen.

Zusätzliche Dienstleistungen umfassen u.a. einen Installationsservice, erweiterten Support, die Entwicklung von Schnittstellen für das System oder auch die Anpassung/Erweiterung/Neuerstellung von Systemfunktionen und -modulen. Kontaktieren Sie uns über http://www.sandoba.de/ und wir machen Ihnen gerne ein unverbindliches Angebot.

Stand: 05/2010, Version: 2010.2